Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Nejc Ribič

Analiza turističnih tokov v mestu na podlagi spletnih objav turistov

diplomsko delo

visokošolski strokovni študijski program
prve stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Damjan Vavpotič
Somentor: izr. prof. dr. Ljubica Knežević Cvelbar

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V turizmu imajo ocene in izkušnje, ki si jih turisti medsebojno izmenjujejo z uporabo različnih turističnih spletnih ali mobilnih aplikacij vedno večji pomen. Iz oddanih mnenj in ocen pa je med drugim mogoče razbrati tudi katere turistične lokacije turisti obiskujejo oz. kakšne so njihove tipične poti. To predstavlja pomembno informacijo za management v turističnih podjetjih in ustanovah. V okviru diplomskega dela razvijte delujoč prototip sistema, ki bo omogočil analizo turističnih tokov v mestu na podlagi spletnih objav turistov. Pri izdelavi diplomskega dela temeljite na obstoječem pristopu za identifikacijo turističnih tokov med destinacijami in ga ustrezno prilagodite za potrebe analize na ravni mesta. Delovanje prototipa preizkusite v praksi in rezultate kritično ovrednotite.

Iskreno se zahvaljujem mentorju izr. prof. dr. Damjanu Vavpotiču, ki me je navdušil nad diplomsko nalogo in me ves čas vodil, spremljal in pomagal reševati kompleksnejše probleme, ki so vodili do samega diplomskega dela.

Zahvaljujem se tudi somentorici izr. prof. dr. Ljubici Knežević Cvelbar, ki mi je pomagala in dodala težo samim rezultatom analize diplomske naloge.

Zahvalil bi se tudi družini, punci, ter vsem drugim, ki so me ves čas tekom študija podpirali, mi stali ob strani in niso obupali nad mano.

Posebna zahvala pa gre tudi Emi Trlep, ki si je vzela čas in lektorirala celotno diplomsko delo.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
CSS	cascade style sheets	kaskadne stilske podloge
RDBMS	relational database management system	relacijski sistem za upravljanje podatkovnih baz
HTML	hyper text markup language	jezik za označevanje nadbesedila
CSV	comma separated values	vejično ločene vrednosti
JSON	JavaScript object notation	JavaScript notacija objekta
URL	uniform resource locator	enolični krajevnik vira
API	application programming interface	aplikacijski programski vmesnik
REST	representational state transfer	prenos reprezentančnega stanja
SSH	secure shell	varna lupina
FTP	file transfer protocol	protokol za prenos datotek

Povzetek

Naslov: Analiza turističnih tokov v mestu na podlagi spletnih objav turistov

Avtor: Nejc Ribič

Turistične spletne aplikacije kot so TripAdvisor zbirajo in prikazujejo objave ter komentarje turistov o različnih lokacijah. Z današnjimi tehnologijami lahko te podatke enostavno zajamemo s spleta. Cilj oziroma problem diplomske naloge je identifikacija in analiza turističnih tokov na podlagi zgoraj omenjenih objav in komentarjev turistov na območju velikosti mesta. Kot rešitev te težave smo izdelali prototip spletne aplikacije, s katero lahko analiziramo turistične tokove, jih vizualno bolje razumemo ter lažje interpretiramo. Turistični tok je ponovljivo gibanje turistov v geografskem prostoru. Za analizo smo zbrali ter analizirali podatke za mesti Ljubljana in Dunaj. Posamezno mesto smo s pomočjo prototipa tudi analizirali in na koncu pridobljene rezultate podrobneje predstavili.

Ključne besede: objave turistov, turistični tok, statistično orodje, vizualizacija poti, analiza območja, spletna aplikacija, ekonomija poti.

Abstract

Title: Analysis of tourism flows in the city based on tourists' online posts

Author: Nejc Ribič

Tourist web applications like TripAdvisor are collecting and displaying posts and comments from tourists on various locations. With today's technologies, we can easily scrap this information from the web. The goal or problem of the thesis is the identification and analysis of tourism flows based on the above mentioned posts, and comments of tourists in the area of the city size. As a solution to this problem, we created a web application prototype, which enables us to analyse tourism flows, to visualize them and to interpret them more easily. The tourism flow is a repeatable movement of tourists in the geographical area. For the analysis we collected and analysed the data for the cities of Ljubljana and Vienna. We analysed the individual city with the help of the prototype and in the end the results we obtained were presented in detail.

Keywords: tourist posts, tourism flow, statistical tool, path visualization, area analysis, web application, path economy.

Poglavje 1

Uvod

Današnje tehnologije imajo močan vpliv na turizem. Omogočajo shranjevanje in serviranje turistu relevantnih informacij. Večina turistov se pred potovanjem zagotovo informira na spletu, kjer dobijo informacije s prve roke. Velik vpliv na odločitev ali bo turist obiskal lokacijo znotraj destinacije ali ne, imajo pozitivni ali bodisi negativni odzivi turistov, ki so že obiskali določeno lokacijo znotraj destinacije.

Današnje turistične aplikacije turista opomnijo in ga povabijo k oddaji subjektivnega mnenja o lokaciji, na kateri se trenutno nahaja oziroma se je nahajal ne dolgo nazaj. Med drugim pa te iste aplikacije turistu tudi predlagajo, kako bi lahko svoj izlet nadaljeval. Prav takšen način oddajanja mnenj posameznih turistov je dober vir podatkov in informacij, ki omogoča različne analize. V okviru dela smo se osredotočili na analiziranje gibanja turistov na podlagi oddanih mnenj.

Zaporedje oddanih mnenj oziroma komentarjev posameznih turistov, bi teoretično lahko interpretirali kot zaporedje dejansko obiskanih turističnih lokacij znotraj destinacij. Z izbiro časovnega okvirja posamezne poti, bi lahko za posameznega turista, iz vseh njegovih objav, sestavili več različnih poti. V primeru, da bi neko pot zaznali pri več turistih - kar bi pomenilo, da je več turistov objavilo komentarje za iste lokacije v enakem vrstnem redu - bi takrat lahko govorili o turističnem toku. Močnejši kot bi turistični tokovi

bili, večja verjetnost bi bila, da turisti lokacije, ki sestavljajo turistični tok, dejansko obiščejo v takšnem vrstnem redu.

Cilj diplomske naloge je priprava postopka in prototipov programskih orodij, ki bi omogočili izvedbo analize turističnih tokov v mestu, na podlagi objavljenih mnenj turistov na turističnem portalu TripAdvisor. Turistični tok v osnovi predstavlja ponovljivo gibanje turistov v nekem geografskem prostoru. Za identifikacijo slednjega potrebujemo ustrezno metodologijo analize ter tri glavne vrednosti v podatkih, nad katerimi izvajamo analizo. To so: unikatni identifikator turista, datum objave komentarja ter koordinate lokacije, za katero je turist podal svoje mnenje oziroma komentar. Pomeben del analize sta predvsem tem večja količina podatkov in čim daljši časovni spekter le-teh. V našem primeru smo izvedli analizo Dunaja in Ljubljane med letoma 2005 in 2018.

Z razumevanjem turistčnih tokov in uporabo statistike, bi mogoče lahko napovedali kje in kdaj bo večji delež turistov ter kako se bodo v prihodnje premikali. S tem bi lahko pripomogli k reševanju problemov o prenatrpanosti glavnih mest v času, ko imajo največ turističnih obiskov. Analizo bi prav tako lahko uporabili za namene ekonomskega ali pa morda celo logističnega planiranja; hkrati pa tudi za namene promocije, trženja in strateškega načrtovanja razvoja.

V diplomskem delu so najprej predstavljena in opisana vsa orodja ter tehnologije, ki so bile uprabljene za razvoj prototipa in so pripomogle k izvedbi analize. Sledi uvodni del, kjer je predstavljen koncept prototipa. Nato je v sklopu glavnega dela predstavljen način spletnega luščenja podatkov, sledi predstavitev uporabljene metodologije analize, kjer so opisani posamezni koraki za detekcijo turističnih tokov. Sledi predstavitev prototipa ter njegove arhitekturne zgradbe, pri čemer je prikazan tudi primer delovanja prototipa. Kot zaključek glavnega dela pa so predstavljeni še rezultati analize, ki smo jo izvedli nad mestoma Ljubljana in Dunaj. V zadnjem delu je predstavljen še sklep ter predlagane možnosti izboljšave prototipa oziroma koncepta.

Poglavje 2

Razvojna orodja in tehnologije

Za zajem podatkov in izdelavo prototipa smo uporabili več različnih orodij in tehnologij. Naš prototip je v osnovi spletna aplikacija, zato smo za razvoj uporabili pogosto uporabljene spletne tehnologije. Pri izbiri tehnologij nas je omejevalo predvsem zakupljeno gostovanje. Izbrati smo morali tehnologije, ki imajo podporo pri našem ponudniku gostovanja. Pri izbiri orodij za zajem podatkov, pa smo iskali take, ki na najlažji in najhitrejši način zajamejo za analizo potrebne podatke. V nadaljevanju so predstavljena in opisana vsa orodja in tehnologije, ki smo jih uporabili pri implementaciji prototipa.

2.1 Tehnologije

2.1.1 JavaScript

JavaScript je objektni skriptni programski jezik, ki je v splošnem namenjen sodelovanju s HTML-kodo. Glavna lastnost JavaScripta je omogočanje interaktivnosti spletnim stranem, ki so pomemben del današnjih spletnih aplikacij. JavaScript je odprtokodni programski jezik, kar pomeni, da za njegovo uporabo ne potrebujemo nobene licence. Programski jezik je razvilo podjetje Netscape leta 1995. Takrat JavaScript še ni imel uradnih standardov, kar pomeni, da so ga različni brskalniki uporabljali in interpretirali različno. Trenutno je uveljavljenih že veliko uradnih standardov, kot so ECMAScript

5 - uveljavljen leta 2009 ter ECMAScript 6 - uveljavljen leta 2015 [6]. V naši aplikaciji uporabljamo JavaScript in na njegovi osnovi zgrajene knjižnice, kot je jQuery za vizualizacijo turističnih tokov ter analizo podatkov.

2.1.2 PHP

PHP je odprtokodni programski jezik in je v osnovi namenjen razvoju dinamičnih spletnih aplikacij [17]. Uporablja se ga za razvoj strežniških aplikacij, kar pomeni, da se program izvede na strani gostitelja, rezultate pa nato prikaže odjemalcu. PHP je skriptni jezik, zato za svoje delovanje potrebuje PHP interpreter. Trenutna stabilna verzija je PHP 7.2.5, ki je bila izdana aprila letos [12]. Ta verzija ima veliko izboljšav v smeri hitrejšega izvajanja. V našem prototipu uporabljamo programski jezik PHP za komunikacijo s podatkovno bazo ter analizo turističnih tokov, na podlagi zbranih podatkov.

2.1.3 MySQL

MySQL je izredno hiter, robusten in relacijsko usmerjen sistem za upravljanje s podatkovnimi bazami [8]. Omogoča hitro in učinkovito izvajanje kompleksnih poizvedb, hkrati pa izredno dobro komunicira s tehnologijama PHP in Apache. Je odprtokodni poizvedovalni jezik, ki za izvajanje poizvedb uporablja programski jezik SQL (angl. Structured Query Language). V spletni aplikaciji ga uporabljamo za shranjevanje podatkov o turističnih točkah, turistih ter njihovih objavah.

2.1.4 HTML, CSS in Sass

HTML je označevalni jezik za izdelavo osnovne strukture spletnih strani s HTML elementi. Ti elementi so predstavljeni z značkami, katere se določa s špičastimi oklepaji [5]. Elementi se lahko med seboj gnezdijo, veljati pa mora, da ima začetna značka tudi svojo zaključevalno značko. Posamezne elemente se lahko poljubno oblikuje s CSS podlogami (angl. cascade style sheets). CSS podloge so namenjene vizualnem oblikovanju spletne strani, kot so barve,

določanju oblike pisave, določanju velikosti posameznih elementov in še veliko drugim nastavitvam. Sass pa je prevajalski skriptni jezik, ki se prevede v jezik CSS [13]. Namenjen je lažjemu in bolj preglednemu programiranju CSS podlog. V naši spletni aplikaciji uporabljamo HTML kot strukturo in postavitev elementov na spletni strani, Sass pa uporabljamo za določanje vizualne oblike posameznih elementov in komponent.

2.2 Razvojna orodja

2.2.1 PhpStorm

PhpStorm je integrirano razvojno okolje za razvoj spletnih aplikacij. Razvito je na osnovi platforme IntelliJ IDEA, katero je razvilo podjetje JetBrains [7]. PhpStorm nudi razvijalcu tekstovni urejevalnik za PHP, HTML, JavaScript in še mnogo drugim tehnologijam. Orodje navdušuje z analizo napisane programske kode in razvijalca že med pisanjem opozarja na napake, še preden se te zares zgodijo. S tem olajša iskanje napak. Ponuja tudi možnost prilagoditve barvne teme in načina postavitve posameznih komponent. Velika dodana vrednost orodja je tudi samo dokončanje (angl. auto complete) programske kode, zaradi česar razvoj postane še hitrejši.

2.2.2 cPanel

cPanel je programska oprema za spletno gostovanje, ki temelji na operacijskem sistemu Linux. Orodje je nameščeno na spletnem strežniku. Uporaba orodja je izredno preprosta, saj ima ogromno funkcionalnosti za administracijo spletne strani [11]. Nadzorna plošča nudi grafični vmesnik, zaradi katerega je samo vzdrževanje spletne strani še bolj pregledno. Enake funkcionalnosti lahko dosežemo tudi preko ukaznega okna, ki ga cPanel ponuja. Do ukaznega okna lahko dostopamo preko protokola SSH. Tehnologije, ki jih med drugim orodje podpira so Apache, PHP, MySQL in druge. Orodje omogoča upravljanje uporabniških FTP-računov, nudi podporo e-poštnem

nabiralniku, omogoča vzdrževanje in dodajanje poddomen ter vzdrževanje in pregled nad podatkovnimi bazami. Nudi še ogromno drugih funkcionalnosti ter možnosti.

2.2.3 phpMyAdmin

phpMyAdmin je odprtokodna brezplačna programska oprema napisana v programskem jeziku PHP. Orodje je namenjeno administraciji podatkovne baze MySQL na spletni strani in je vodilno na tem področju [4]. Novejše različice orodja nudijo podporo tudi sistemu MariaDB. Orodje se izvaja na strežniku in je posledično neodvisno od operacijskega sistema, ki ga ima uporabnik. Osnovne funkcionalnosti, ki jih orodje ima, so dodajanje, urejanje, spreminjanje in prikazovanje podatkov v podatkovni bazi. Vsebuje tudi naprednejše funkcionalnosti, kot so ustvarjanje podatkovnih baz, kontrolo uporabnikov, dodeljevanje pravic uporabnikom, manipulacijo s trigerji, indeksi in še veliko drugih možnosti. Največja prednost orodja pa je enostaven in pregleden uporabniški vmesnik.

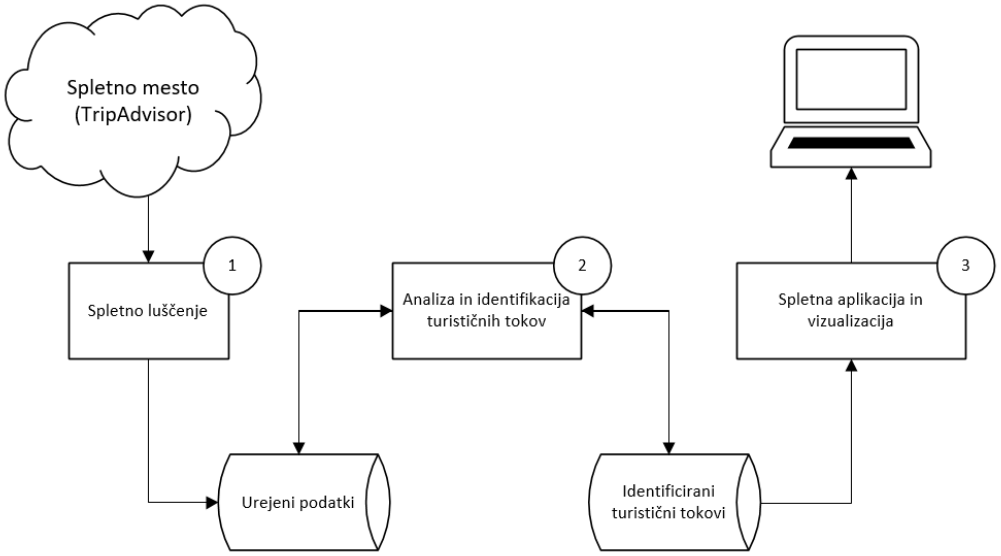
Poglavje 3

Konceptualna zgradba prototipa

V tem poglavju je predstavljen in opisan osnoven koncept prototipa. Prikazano je zaporedje in sodelovanje med glavnimi komponentami, ki se izvajajo v procesu analize in vizualizacije turističnih tokov. Omenjeno zaporedje je prikazano tudi s shemo. V splošnem je to poglavje namenjeno obrazložitvi širše slike prototipa, ki smo ga izdelali za analizo turističnih tokov.

3.1 Zgradba prototipa

Za analizo in identifikcijo čim močnejših turističnih tokov posameznega mesta, potrebujemo čim večjo množico podatkov. Poleg podatkov pa potrebujemo tudi ustrezno metodologijo analize nad podatki, ki dejansko identificira turistične tokove. V osnovi metodologija o identifikaciji turističnih tokov izhaja iz članka, kjer je predstavljena analiza turističnih tokov v Slovenji [3]. Navsezadnje pa potrebujemo za lažje ter boljše razumevanje turističnih tokov tudi vizualizacijo. Na sliki 3.1, so kot oštevilčeni koraki prikazane zgoraj omenenje komponente.



Slika 3.1: Koraki izvedbe analize turističnih tokov.

Spletno luščenje

Kot prvi korak analize turističnih tokov, potrebujemo čim večjo zbirko oziroma množico ustreznih podatkov. Slednjo smo v potrebni obliki (za našo analizo) zasledili na spletnem mestu TripAdvisor. Najlažji način pridobivanja podatkov s spletnih mest je tako imenovano spletno luščenje (angl. web scraping). Na sliki 3.1 je korak označen s številko *1*. Po končanem luščenju je pridobljene podatke smiselno programsko preveriti in po potrebi tudi urediti z namemom, da se izognemo napačni interpretaciji rezultatov analize. Zaradi ogoromne količine podatkov in časa, ki ga porabimo za spletno luščenje, je pridobljene podatke smiselno hraniti v podatkovni bazi. Sam koncept ter proces luščenja, kot tudi urejanja podatkov, je predstavljen v poglavju 4.

Analiza in identifikacija tokov

Ko so podatki shranjeni in urejeni, sledi korak analize in identifikacije turističnih tokov. Korak je na sliki 3.1 označen z zaporedno številko *2*. Za identifikacijo tokov prototip uporabi specifično metodologijo nad podatki ter

s tem identificira turistične tokove. V osnovi metodologija za vsakega turista sestavi eno daljšo pot (sestavljeno iz vseh njegovih objav), ki jo nato razdeli na več krajših poti. Na koncu poišče še unikatne poti, katerim določi število ponovitev. Uporabljena metodologija je podrobneje predstavljena v poglavju 5. Rezultate analize, tj. identificirane turistične tokove, prototip nato preko aplikacijskega vmesnika nudi odjemalcu.

Spletna aplikacija in vizualizacija

Kot zadnji korak pa sledi korak vizualizacije, ki je na sliki 3.1 označen pod številko *3*. Vizualizacija je sestavni del spletne aplikacije, ki jo vidi uporabnik. V osnovi jo sestavljajo kontrolna plošča, kjer uporabnik določi parametre in zahteve za identifikacijo turističnih tokov, ter zemljevid, kjer se izvede končna vizualizacija turističnih tokov. Turistične tokove spletna aplikacija prejme iz koraka *2*. Vizualizacija omogoča bolj pregledno analizo in medsebojno primerjavo različnih turističnih tokov. Pristop k vizualizaciji je predstavljen v poglavju 6, kjer predstavimo prototip.

Poglavje 4

Spletno luščenje podatkov

Današnja spletna mesta zbirajo in hranijo ogromno količino podatkov. Ti podatki se v večini primerov hranijo v podatkovnih bazah, ki pa niso dostopne navadnim uporabnikom, temveč zgolj skrbnikom njihovih spletnih mest. Obstaja več načinov kako lahko uporabnik dostopa do podatkov. Eden izmed najbolj preprostih načinov je preko aplikacjiskega vmesnika, ki ga ponuja to spletno mesto, na žalost pa so le-ti pogosto močno omejeni. Drug način dostopa do podatkov pa je s pomočjo spletnega luščenja podatkov. V tem primeru je uporabnik omejen zgolj na podatke, ki jih vidi s pomočjo spletnega brskalnika na spletni strani. Navsezadnje pa so spletne strani predstavljene zgolj v tekstovni obliki, v tako imenovanem HTML formatu, ki služi kot način prikazovanja vsebine spletne strani v spletnem brskalniku. Več o HTML formatu je predstavljeno v podpoglavju 2.1.4.

Spletno luščenje podatkov v osnovi deluje tako, da skripta za luščenje (angl. scraper), ki zbira podatke, najprej pošlje GET-poizvedbo do željene spletne strani ter prenese njen izvorni HTML-dokument. Nato skripta po HTML-dokumentu preišče, zbere in uredi željene podatke ter jih shrani v željeni format. Obstaja ogromno knjižnic, ki ta proces opravijo namesto nas. Obstajajo pa tudi vtičniki za spletne brskalnike, ki simulirajo naše klikanje po spletni strani, hkrati pa zbirajo željene podatke in nam jih po izvajanju vrnejo v željenem formatu. Glavna slabost spletnega luščenja podatkov je

stalno posodabljanje ter spreminjanje spletnih strani. Pot, ki smo jo uporabili za dostop do podatkov v HTML-dokumentu bo lahko ob naslednjem luščenju drugačna in zgodi se lahko, da nevede izluščimo neveljavne in za nas neuporabne podatke.

V tem poglavju je predstavljen pristop k luščenju podatkov, ki smo ga uporabili za zajem podatkov o Dunaju ter Ljubljani.

4.1 Uporabljen pristop zajema in orodja

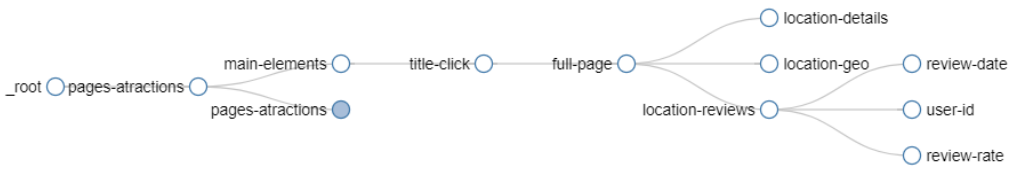
Podatke, ki smo jih potrebovali za analizo, smo zajeli s spletne turistične platforme TripAdvisor. Ker v diplomskem delu analiziramo turistične tokove na podlagi spletnih objav turistov, so ravno objave temeljni podatek, ki ga potrebujemo. Poleg objav pa v sklopu naše analize potrebujemo tudi podatke o posameznih turistih. Zaradi ločenega prikazovanja objav in podatkov o turistih, na spletnem mestu TripAdvisor, smo se zajema lotili v dveh korakih. Prvi korak je bil zajem vseh objavljenih komentarjev za izbrani kraj. V našem primeru je to pomenilo zajem komentarjev za Ljubljano in Dunaj. To smo dosegli s pomočjo vtičnika za brskalnik z imenom WebScraper. Podrobnejše opisana metoda zajema in delovanje vtičnika je predstavljeno v podpoglavju 4.1.1. Drugi korak pa je bil zajem vseh meta podatkov o turistih, kot so starost, spol, tip turista in podobno. Tu nas je vtičnik omejeval, zato smo uporabili programsko skripto. Njeno delovanje in pristop zajema je predstavljeno v podpoglavju 4.1.2. Zbrane podatke smo na koncu preverili, jih uredili in si s tem zagotovili konsistentnost pri shranjevanju v podatkovno bazo ter za kasnejšo analizo.

4.1.1 Zajem komentarjev s spletnim vtičnikom

Prvi korak, ki smo ga za zajem podatkov uporabili, je zajem vseh objav za različne lokacije. To smo storili s spletnim vtičnikom WebScraper. WebScraper je vtičnik za spletni brskalnik Chrome in je namenjen spletnemu zajemu podatkov. Vtičnik je razvilo podjetje Web Scraper, ki je specializirano za

zajem podatkov s spleta.

Pred zajemom smo vtičniku najprej določili elemente, ki za naš zajem pridejo v poštev. To smo storili s preprostim klikom na željeno besedilo ali element. Izbranemu elementu smo hkrati določili še podatkovni tip in čas, ki mora preteči, da vtičnik izlušči in shrani izbrani podatek. Izbira časa je bila tudi največja težava s katero smo se soočili pri zajemu z vtičnikom. Če se v izbranem času komponentna ni prikazala v celoti, je takrat vtičnik zajel napačen podatek, kar je posledično pomenilo, da smo bili pripomorani ponovno začeti z zajemom. Težavo smo rešili tako, da smo čas prikazovanja posamezne komponente določili glede na povprečno vrednost, ki smo jo pridobili s preizkušanjem. Na koncu smo vtičniku določili še konfiguracijsko datoteko v formatu JSON, ki določa strukturo sprehajanja ter vrstni red zajema podatkov. Primer grafa strukture za zajem z vtičnikom je prikazan na sliki 4.1. Na njej lahko vidimo besedilo ter točke. Vsaka točka je lahko bodisi



Slika 4.1: Graf strukture zajema podatkov.

polna bodisi prazna. V našem primeru je polna točka *pages-attractions*, ki predstavlja rekurzivno sprehajanje po straneh in referencira na prazno točko z enakim imenom (*pages-attractions*). Prazne točke v končnih vozliščih pa v splošnem predstavljajo končni podatek, ki ga vtičnik zajame. Ko smo začeli z zajemom, se je odprla nova instanca brskalnika Chrome, ki avtomatsko odpira željene strani in zbira ustrezne podatke, v našem primeru komentarje. Podatke smo po koncu izvajanja prenesli v datoteki, v formatu CSV.

4.1.2 Zajem podatkov o turistih

Iz množice komentarjev, ki smo jih pridobili kot rezultat koraka 4.1.1, smo najprej odstranili vse duplikate turistov in nato za unikatne zajeli njihove meta podatke. Za slednje smo uporabili programsko opremo Scrapy. Scrapy je brezplačna in odprtokodna programska oprema, ki nudi ogrodje za spletno luščenje podatkov in je napisana v programskem jeziku Python [14]. Pred zajemom smo programski opremi določili nekaj glavnih konfiguracij, kot so `ROBOTSTXT_OBEY`, ki določa ali bo robot upošteval pravila spletnega mesta, in `DOWNLOAD_DELAY`, ki določa na koliko časa bo robot ali pajek poslal poizvedbo GET na spletno mesto. Ta pravila so ponavadi zapisana v standardu za izključitev robotov (angl. robots exclusion standard), shranjena pa so v datoteki `robots.txt`, ki se nahaja v korenu spletnega mesta. V sklopu našega zajema se je pojavila težava, ker v omenjenih pravilih ni bilo specificirane vrednosti, ki določa na koliko časa naj robot oziroma pajek pošlje poizvedbo GET na spletno mesto. Kot rešitev smo konfiguracijo skripte nastavili tako, da je spletni pajek po vsaki poizvedbi počakal eno sekundo. S tem smo zmanjšali verjetnost detekcije mehanizmov, ki zaznavajo spletne pajke in luščilce na spletnem mestu TripAdvisor.

Za vsakega turista je skripta prenesla HTML-dokument z njegovimi meta podatki. Podatke, ki so bili za našo analizo relevantni, smo v dokumentu HTML alocirali s poizvedovalnim jezikom XPath, v programskem jeziku Python. Preprost primer zajema podatka *user_id* je prikazan na sliki 4.2. Po končanem luščenju smo podatke shranili v datoteko, v formatu CSV.

```
#Primer poizvedbe
>>> response = scrapy.Request(url=start_url)

# Luščenje vrednosti user_id
>>> response.xpath('//user_id')
[<Selector xpath='//user_id' data='<p>A757BC4B</p>'>]

>>> response.xpath('//user_id/text()').extract_first()
'A757BC4B'
```

Slika 4.2: Primer poizvedbe v poizvedovalnem jeziku XPath.

4.1.3 Pregled podatkov

Kot zadnji korak spletnega luščenja, sledi korak pregleda podatkov. Zaradi pogostega posodabljanja spletnih strani in spreminjanja njihovega izgleda se lahko zgodi, da so po koncu zajema naši podatki neuporabni. Z namenom, da bi pravočasno zaznali neustrezno zajete podatke, smo napisali programsko skripto, ki se sprehodi preko vseh podatkov ter bodisi odstrani bodisi popravi vse neveljavne vrednosti. Neveljavne vrednosti so na primer: vrednost null, drugačen podatkovni tip kot ga pričakujemo, neveljavna struktura podatkov in še veliko drugih. V splošnem nam skripta uredi podatke in poskrbi za njihovo konsistentnost. Na ta način si zagotovimo, da lahko podatke vnesemo v podatkovno bazo in nad urejenimi podatki izvajamo analize.

Poglavje 5

Uporabljena metodologija analize podatkov

Metodologija predstavljena v tem poglavju izhaja iz članka, v katerem je izvedena analiza turističnih tokov v Sloveniji [3]. Slednjo metodologijo smo za naše potrebe preuredili in jo izboljšali tako, da lahko analizo izvajamo na nivoju mesta oziroma posameznih lokacij. Osnovnemu pristopu metodologije smo dodali tudi atribute o turistih in s tem izvedli še analizo turistov na turističnih tokovih.

Podatke uporabljene v analizi turističnih tokov, smo zajeli s turističnega portala TripAdvisor. V sklopu diplomske naloge smo izvedli analizo turističnih tokov nad mestoma Ljubljana in Dunaj. Analiza Ljubljane je vključevala nekaj manj kot 70000 komentarjev, s strani približno 32500 različnih uporabnikov, na 566 različnih lokacijah. Analiza na območju Dunaja pa je vključevala nekaj več kot 220000 komentarjev, s strani približno 80000 različnih uporabnikov, na 555 različnih lokacijah. Razlog, da ima Ljubljana več lokacij kot Dunaj je v tem, ker smo za Ljubljano, zaradi manjšega števila objav, dodali tudi podatke oziroma komentarje o restavracijah, pri Dunaju pa smo upoštevali le podatke o atrakcijah. Podatki Ljubljane ter Dunaja so pridobljeni v časovnem obdobju od začetka leta 2005 do začetka leta 2018.

Izmed vseh uporabnikov je tako za mesto Ljubljana, kot tudi za me-

sto Dunaj v povprečju 31% ljudi poročalo o njihovem spolu, od tega 53% uporabnikov moškega spola ter 47% ženskega. O starosti je poročalo 29% uporabnikov. Od tega 9% starejših od 65 let, 31% starih med 50 in 64 let, 34% starih med 35 in 49 let, 21% starih med 25 in 34 let, 4% starih med 18 in 24 let, manj kot 1% pa starih med 13 in 17 let.

Pri analizi turističnih tokov je najbolj pomembno, da podatki vsebujejo tri pomembne vrednosti. To so koordinate lokacije, čas objavljenega komentarja in enolični identifikator uporabnika. Glavni namen teh vrednosti je izgradnja turističnega toka, ki je podrobno ter po korakih predstavljen v podpoglavju 5.2.

5.1 Opis podatkov

Podatkovni model se po zajemu v splošnem deli na dve glavni tabeli. To sta tabela, ki vsebuje seznam komentarjev ter tabela, ki vsebuje seznam vseh turistov. Obe tabeli vsebujeta atribut, ki ima lahko več vrednosti. Posledično nobena od tabel ne zadostuje 1. normalni obliki. Razlog, da tabel nismo normalizirali je ta, da smo s tem pripomogli k hitrejšem procesiranju poizvedb ter identifikaciji turističnih tokov. Tabela, ki vsebuje seznam vseh komentarjev, je v našem podatkovnem modelu imenovana *Post* in je v relacijski shemi predstavljena na sledeč način:

Post(*review_id* : number, *place_name* : string, *place_details* : string,
lat : decimal, *lng* : decimal, *review_date* : number, *review_rate* : number,
user_id : string)

Atribut *review_id* enolično določa posamezen komentar. Ta atribut je pomemben predvsem zaradi njegove avtomatske inkrementalnosti, ki jo kasneje upoštevamo pri določanju vrstnega reda obiskov lokacij pri izgradnji poti v podpoglavju 5.2.2. Vsak komentar vsebuje tudi vse podatke o lokaciji, ki jo je turist komentiral. V atributu *place_name* hranimo torej ime pod katerim je lokacija predstavljena. Za boljšo analizo pa vsebujejo naši podatki tudi

več vrednostni atribut *place_details*, v katerem so z vejico ločene posamezne podrobnosti o lokaciji. S tem atributom lahko pri analizi turističnih tokov lažje ugotovimo ali nek turist raje obiskuje parke, znamenitosti ali pa morda celo restavracije. Atributa *lat* in *lng* predstaljata geografsko lokacijo in imata decimalno vrednost. Poleg lokacij je v sklopu komentarja vsebovan tudi referenčni podatek o turistu, ki je komentar objavil. Omenjeni podatek hranimo v atributu *user_id*, ki enolično določa posameznega turista. Vsak turist ob objavi komentarja odda svojo subjektivno oceno lokacije, ki jo hranimo v atributu z imenom *review_rate*, datum objavljenega komentarja pa hranimo v atributu *review_date*.

Podatke o posameznih turistih hranimo v tabeli imenovani *Tourist*. Njena relacijska shema je predstavljena na sledeč način:

Tourist(*user_id* : string, *travel_style* : string, *age* : string, *gender* : string)

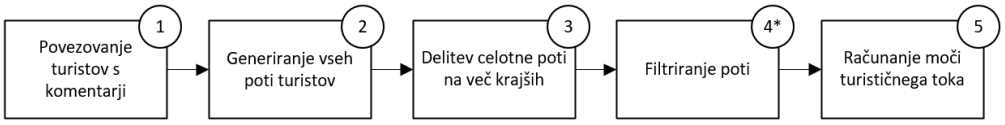
Atribut *user_id* enolično določa posameznega turista. Za slednjega v večvrednostnem atributu *travel_style* hranimo njegov način potovanja. V atributu so z vejico ločene lastnosti načina potovanja turista. S pomočjo tega atributa v analizi, pri močnejših turističnih tokovih, ugotovimo kakšen tip turistov prepotuje to pot in jo hkrati s tem razlogom tudi lažje interpretiramo. Za vsakega turista hranimo tudi njegove osnovne oziroma meta podatke, kot sta spol (*gender*) in starost (*age*).

5.2 Identifikacija turističnega toka

Turistični tok je ponovoljivo gibanje turistov v geografskem prostoru. Tok lahko vsebuje več turističnih točk, vendar morajo biti le-te urejene v kronološkem vrstnem redu. V diplomski nalogi smo komentar, ki ga je objavil turist v povezavi z neko lokacijo (Post), interpretirali kot obisk te lokacije. Kronološko zaporedje obiskov lokacij smo za turista interpretirali iz njegovega kronološkega zaporedja oddajanja komentarjev in na ta način zgradili njegovo pot. Obstaja možnost, da turist teh lokacij ni obiskal v takšnem

zaporedju, vendar v sklopu diplomske naloge analiziramo zgolj močnejše turistične tokove. Moč turističnega toka narašča, če več turistov odda objave iz istih lokacij, v enakem vrstnem redu. Pri oblikovanju pristopa smo se zgledovali po članku [3], ki pa za razliko od našega pristopa ne gradi poti na nivoju posameznih lokacij, ampak na nivoju turističnih destinacij.

Za identifikacijo turističnih tokov je potrebno zajete podatke peljati skozi določene korake. Posamezni koraki so predstavljeni na sliki 5.1 in so podrobneje opisani in predstavljeni v nadaljevanju vsebine poglavja.



Slika 5.1: Koraki identifikacije turističnih tokov.

5.2.1 Povezovanje turistov s komentarji

Za lažjo in bolj pregledno identifikacijo turističnega toka želimo najprej vse podatke, ki jih imamo na voljo, shraniti v zgolj eno relacijo. To naredimo tako, da povežemo relacijo, ki vsebuje seznam vseh komenarjev (Post), z relacijo, ki vsebuje podrobnejše podatke o turistih (Tourist). Poizvedba 5.1, poveže vse turiste z njihovimi obiskanimi lokacijami, rezultat pa nato shrani v novo relacijo z imenom *Trip*.

$$\rho_{Trip}(Tourist \bowtie_{Tourist.user\_id = Post.user\_id} Post) \tag{5.1}$$

Relacija *Trip* ima na koncu vsebovane vse atribute relacije *Post* ter relacije *Tourist*. Vsak posamezni atribut omenjenih tabel pa je že podrobneje opisan v podpoglavju 5.1. Celotna relacijska shema relacije *Trip* je predstavljena na sledeč način:

Trip(*review_id* : number, *user_id* : string, *place_name* : string, *place_details* : string, *lat* : decimal, *lng* : decimal, *travel_style* : string, *age* : string, *gender* : string, *review_date* : number, *review_rate* : decimal)

5.2.2 Generiranje vseh poti turistov

Z združenimi podatki, ki smo jih pridobili kot rezultat poizvedbe 5.1, v naslednjem koraku zgradimo dejansko pot, ki jo je posamezni turist opravil. Kakor je bilo že omenjeno v uvodu tega poglavja, interpretiramo zaporedje oddajanja komentarjev kot dejansko zaporedje obiska. Na podlagi tega, s pomočjo poizvedbe 5.2, zgradimo eno dolgo pot za vsakega turista, ki ima vsebovane vse njegove obiskane lokacije. Rezultat poizvedbe 5.2 nato shranimo v dodatno relacijo (Paths), ki je predstavljena na koncu trenutnega koraka.

$$\begin{aligned}\rho_{Paths}\ (\Pi_{user\_id,\ age,\ gender,\ travel\_style,\ path}\ (\\ \rho_{path}\ (_{user\_id}\ \mathfrak{F}_{group\_concat(}\\ lat,\ ':',\ lng,\ ':',\ place\_name,\ ':',\ place\_details,\ ':',\ review\_rate,\ ':',\ review\_date\\ ORDER\ BY\ user\_id,\ reivew\_id\ ASC\ SEPARATOR\ ';')(Trip)\ )\ )\ )\end{aligned} \tag{5.2}$$

Poizvedba za vsakega turista, katero v našem primeru ločimo z atributom *user_id*, poišče vse njegove objave ter jih uredi naraščajoče po atributu *review_id*. Poizvedba je predstavljena v relacijski algebri. Ta ima na voljo zgolj osnovne poizvedovalne operacije. V naši poizvedbi smo uporabili napredno funkcijo *group_concat()*, ki se jo uporablja v poizvedovalnem jeziku SQL. Rezultat omenjene operacije grupiranja združi vse ustrezne atribute v en zaporeden niz, ki je urejen naraščajoče po atributu *user_id* ter *review_id*. V našem primeru smo posamezne atribute v omenjenem nizu ločili z znakom ":", posamezne obiske lokacij pa z znakom ";". Takšen način ločevanja grupiranih atributov in lokacij, nam v naslednjem koraku, torej koraku 5.2.3, olajša delitev glavne poti na več manjših. Po operaciji grupiranja se lahko

zgodi, da ima posamezna pot vsebovane tudi dve ali več zaporednih objav iz iste lokacije. Način interpretacije takšne poti je predstavljen nekoliko kasneje. Relacijska shema relacije (Paths), kjer so shranjeni rezultati poizvedbe 5.2, je predstavljena na sledeč način:

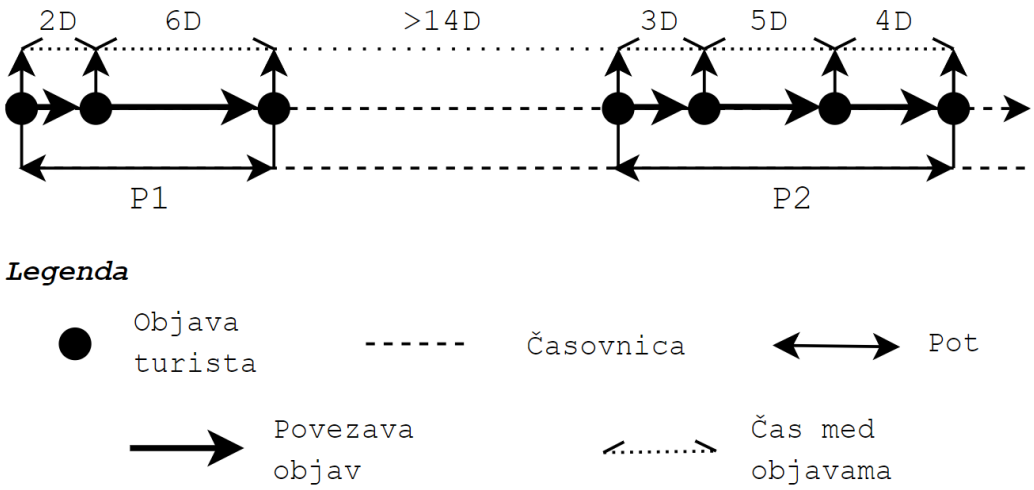
$$\text{Paths}(\underline{id : \text{number}}, user\_id : \text{string}, age : \text{string}, gender : \text{string}, travel\_style : \text{string}, path : \text{string})$$

V relaciji sta dodana dva nova atributa. Dodan je atribut *id*, ki ima številčno vrednost ter lastnost avtomatske inkrementalnosti. Dodan je tudi atribut *path*, ki je rezultat operacije grupiranja (omenjena zgoraj), vsi ostali atributi pa so predstavljeni že v podpoglavju 5.1.

5.2.3 Delitev celotne poti na več poti

V prejšnjem koraku, v poizvedbi 5.2 je rezultat za vsakega turista zgolj ena pot (*path*). Upoštevajoč naše uporabljene podatke je to pot, ki jo je opravil turist v izbranem kraju od začetka leta 2005 do začetka leta 2018. Takšna interpretacija poti seveda z analitičnega stališča ni ustrezna, zato pot naknadno s pomočjo atributa *review_date* - atribut je grupiran znotraj niza v atributu *path* - razdelimo na več krajših poti. Pot namreč delimo, ker sklepamo, da gre za isto pot dokler so posamezna poročanja turistov dovolj skupaj. Enako sklepanje je predstavljeno v članku [3]. Metodo, ki smo jo v sklopu deljenja poti uporabili, je predstavljena na sliki 5.2. Z omenjeno metodo izhajamo iz metode deljenja poti, ki je pravtako predstavljena v članku [3]; vendar je njihov koncept iskanja krajših poti povsem drugačen, kot ga uporabimo in predstavimo mi. Prednosti našega pristopa, so učinkovistost in hitrost izvajanja analize ter predvsem lažja implementacija rešitve v dejanskem prototipu.

Deljenje poti je potrebno izvesti za vsakega turista posebej. To naredimo tako, da se sprehodimo preko relacije *Paths* za vsakega turista in po potrebi razdelimo njegovo skupno pot, ki je shranjena v atributu *path*. Pred deljenjem poti je najprej potrebno določiti časovno območje, ki bo določalo



Slika 5.2: Metoda deljenja poti na več manjših poti.

dovoljen razmak med objavami. V primeru, ki je prikazan na sliki 5.2, je izbrano časovno območje obsegalo 14 dni. To pomeni, da bo posamezna pot sestavljena iz lokacij, ki jih je turist komentiral in se med seboj ne razlikujejo za več kot to določa izbrano obdobje. Z implementacijskega vidika pa to pomeni, da vsaka nova objava spada v eno pot, če je njena datumska razlika od zadnje objave v tej poti manjša ali enaka od izbranega časovnega območja. Rezultat metode nato shranimo v relacijo *TouristPaths*, katere relacijska shema je enaka relaciji *Paths*. Razlika relacije *TouristPaths* je zgolj v tem, da se posamezni turist lahko pojavi večkrat, v relaciji *Paths* pa se je vedno pojavil le enkrat.

Na sliki 5.2 so razlike posameznih objav označene s številko, ki ji sledi črka *D*. To predstavlja datumsko razliko dveh sosednjih objav (oz. obiskov), ki jo izračunamo s pomočjo grupiranega atributa *review_date*. Obiski, ki jih je posamezni turist opravil in ustrezajo izbranemu časovnemu obdobju, predstavljajo pot, ki je na sliki označena s črko *P (tourist_path)* in njeno zaporedno številko. Pogoj, da pot interpretiramo kot veljavno, sta vsaj dva obiska različnih lokacij v izbranem časovnem obdobju. Pomemben pogoj, ki lahko močno vpliva na rezultate analize je, da v posamezni poti ne smeta

obstajati dva ista zaporedna kraja. Primer poti, kjer sta prisotna dva enaka zaporedna kraja: A-B-B-C-D, v tem primeru je pot potrebno interpretirati kot: A-B-C-D.

Smer poti

V koraku deljenja celotne poti na več manjših je smiselno omeniti, da lahko pot A-B-C prehodimo tudi na način C-B-A. Ta pot je v osnovi enaka, vendar je lahko tip turista oziroma interpretacija te poti drugačna, če upoštevamo njeno obratno smer. V prototipu smo izdelali funkcionalnost združevanja teh poti, ki je predstavljena v podpoglavju 6.2.1. Smer poti je po našem mnenju pametno upoštevati, kadar analiziramo posamezne turiste in preučujemo njihovo obnašanje. Kadar pri analizi opazujemo tokove s stališča povezanega obiskovanja različnih turističnih točk, pa je pogosto smer obhoda mogoče zanemariti, kar nam omogoči, da dobimo večje moči turističnih tokov. Računanje moči turističnega toka je predstavljeno v zadnjem koraku, v podpoglavju 5.2.5.

5.2.4 Filtriranje turističnih tokov *

Ta korak je v splošnem dodaten korak in prikazuje primer filtracije turističnih tokov. Slednje dosežemo tako, da iz relacije *TouristPaths* preberemo zgolj vrstice, ki ustrezajo nekemu izbranemu pogoju. Poizvedba 5.3 prikazuje primer filtracije turističnih tokov za neko izbrano pot.

$$\sigma_{path\ =\ 'selected\_path'}(TouristPaths) \tag{5.3}$$

Rezultat poizvedbe 5.3 so vrstice relacije *TouristPath*, katere imajo pot - shranjena v atributu *path* - enako neki poljubno izbrani poti (*selected_path*). V splošnem to pomeni, da kot rezultat dobimo seznam vseh turistov s potjo, ki so kadarkoli prehodili izbrano pot.

Posamezne poti lahko omejimo tudi glede na lastnosti turista in kraja. Primer takšne filtracije prikazuje poizvedba 5.4.

$$\sigma_{travel\_style\ LIKE\ ('\%tourist\_type\%')\ AND\ path\ LIKE\ ('\%place\_type\%')}\ (TouristPaths) \tag{5.4}$$

V poizvedbi smo uporabili napredno funkcionalnost *LIKE()*, ki preveri vsebovanost podniza v določenem atributu. Pogoj v naši poizvedbi je posledično veljaven takrat, kadar atributa *travel_style* in *path* vsebujeta prednastavljene vrednosti. V splošnem to pomeni, da poizvedba 5.4 vrne seznam vseh poti in turistov, ki imajo vsebovane ustrezne lastnosti.

5.2.5 Računanje moči turističnega toka

V koraku 5.2.3 smo razdelili skupno pot vsakega turista na več krajših glede na prednastavljeno časovno območje ter nato rezultate shranili v relacijo *TouristPaths*. V tem koraku pa izračunamo moč posameznega turističnega toka. Moč izračunamo s pomočjo poizvedbe 5.5.

$$\rho_{TouristFlows}\ (\ \Pi_{path,\ strength}\ (\ \rho_{strength}\ \ (_{path}\ \mathfrak{F}_{count(user\_id)}(TouristPaths))\ )\ ) \tag{5.5}$$

Poizvedba 5.5 za vsako pot prešteje število različnih turistov, ki so to pot prepotovali, ter nato rezultat shrani v atribut *strength*. Podobno poizvedbo za izračun moči uporabijo tudi v članku [3]. Vse turistične tokove ter njihove moči poizvedba nato shrani v relacijo *TouristFlows*, katere relacijska shema je predstavljena na sledeč način:

TouristFlows(*id* : number, *path* : string, *strength* : number)

Relacija *TouristFlows* vsebuje atribut *id*, ki enolično določa posamezen turistični tok. Atribut *path* določa turistični tok, njegova moč pa je določena v atributu *strength*.

Poglavje 6

Predstavitev prototipa

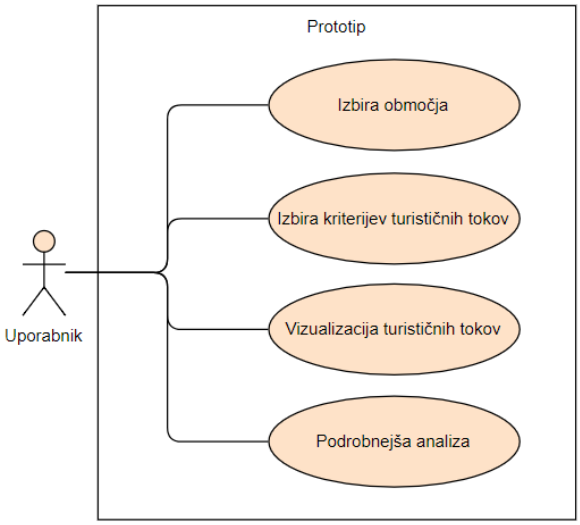
V tem poglavju je podrobneje opisan in predstavljen prototip. Najprej je predstavljena celotna arhitektura prototipa, kjer osnovne funckionalnosti predstavimo s pomočjo diagrama primerov uporabe. Sledi predstavitev strukture podatkov, ki jih hranimo v podatkovni bazi. S pomočjo komponentnega diagrama nato predstavimo glavne komponente ter njihovo medsebojno komunikacijo. Nato s pomočjo diagrama aktivnosti prikažemo primer vizualizacije turističnih tokov, kjer posamezne aktivnosti še posebej obrazložimo. Na koncu sledi še predstavitev prototipa, kjer so predstavljeni posamezni glavni elementi, kot so kontrolna plošča, vizualizacija in komponenta podrobnejše analize turističnega toka ali lokacije. Vsako izmed njih podrobneje opišemo ter podpremo z zaslonsko sliko.

6.1 Arhitektura prototipa

6.1.1 Koncept prototipa

Prototip je v osnovi spletna aplikacija, ki uporabniku omogoča enostavno pregledovanje in analiziranje turističnih tokov. Za lažjo predstavitev operacij, ki jih uporabnik lahko izvaja, smo za prikaz uporabili diagram primerov uporabe. Diagram primerov uporabe predstavlja, na kakšen način je lahko nekdo v interakciji s sistemom. V splošnem ga sestavljajo akter, primeri upo-

rabe ter njihove medsebojne povezave. Akter je nekdo, ki je na kakršenkoli način v interakciji s sistemom, primer uporabe pa predstavlja abstraktno funkcionalnost, ki jo sistem izvaja ali nudi. Interakcijo oziroma razmerje akterja in primera uporabe pa predstavimo s povezavo. Diagram primerov uporabe prototipa je prikazan na sliki 6.1. Akter-uporabnik ima v prototipu



Slika 6.1: Diagram primera uporabe uporabnika.

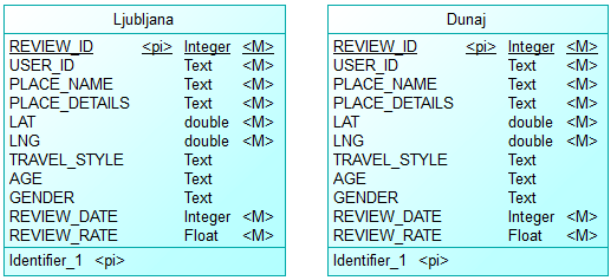
možnost izbire območja (oz. mesta), kjer želi identificirati turistične tokove. Določi in izbere lahko kriterije, ki vplivajo na samo identifikacijo turističnih tokov. Ti kriteriji so: število ponovitev turističnega toka, dolžina turističnega toka, časovno območje posamičnih poti, tip turista ter tip lokacije. Uporabnik lahko izbere tudi časovno obdobje analize, lahko upravlja z vizualizacijo in njenimi nastavitvami, hkrati pa ima tudi možnost podrobnejše analize posameznih turističnih tokov ter točk.

6.1.2 Struktura podatkov

Zaradi ogromne količine podatkov, nad katerimi izvajamo analizo in vizualizacijo, bi bil zajem podatkov v realnem času nemogoč. S tem razlogom vse

potrebne podatke zajamemo že vnaprej in jih hranimo v podatkovni bazi. V podpoglavju 5.1 je predstavljena struktura podatkov po zajemu. V dejanski implementaciji pa se izkaže, da čas izvajanja operacije združevanja (angl. join operation) porabi preveč časa. S tem razlogom korak, ki je predstavljen v podpoglavju 5.2.1, izvedemo že vnaprej in si tako pripravimo zgolj eno tabelo, ki vsebuje že združene podatke o komentarjih, lokacijah ter turistih. V tem istem koraku (korak 5.2.1) je predstavljena tudi sama struktura omenjene tabele.

Nad omenjeno osnovno tabelo, za še hitrejšo analizo, izvedemo horizontalno razbitje na več particij. Poleg indeksov in materializiranih pogledov sta horizontalna in vertikalna delitev particij pomembna vidika oblikovanja relacijskih podatkovnih baz, ki bistveno izboljšajo hitrost obdelave poizvedb [1]. V našem primeru to pomeni, da vsako mesto, nad katerim izvajamo analizo, hranimo v svoji tabeli. Končni podatkovni model, ki ga pri dejanski analizi Ljubljane in Dunaja uporabljamo, je predstavljen na sliki 6.2.

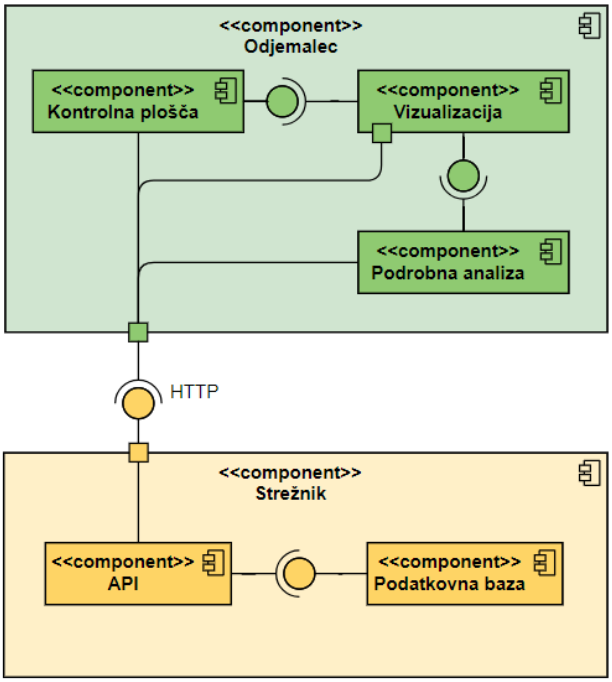


Slika 6.2: Podatkovni model uporabljen v prototipu.

V splošnem vsaka od tabel vsebuje seznam vseh komentarjev, ki so bili v izbranem mestu objavljeni. Entiteti (*Ljubljana* in *Dunaj*) imata kot primarni ključ unikatni identifikator komentarja. Poleg primarnega ključa pa o komentarju hranita tudi podano oceno ter čas objave. Obe entiteti hranita tudi vse podatke o turistih, kot so spol, starost ter enolični identifikator turista. Hkrati vsebujeta tudi vse podatke o lokacijah, njihovih imenih in njihovih podrobnostih.

6.1.3 Zgradba prototipa

Za lažjo predstavitev zgradbe protipa smo uporabili komponentni diagram. Komponentni diagram je formalna tehnika predstavitve kompleksnejših sistemov, kjer so s pomočjo vmesnikov prikazane posamezne povezave med komponentami sistema. Komponentni diagram prototipa je predstavljen na sliki 6.3. Prototip se v osnovi deli na dva dela, to sta strežnik in odjema-



Slika 6.3: Komponentni diagram prototipa.

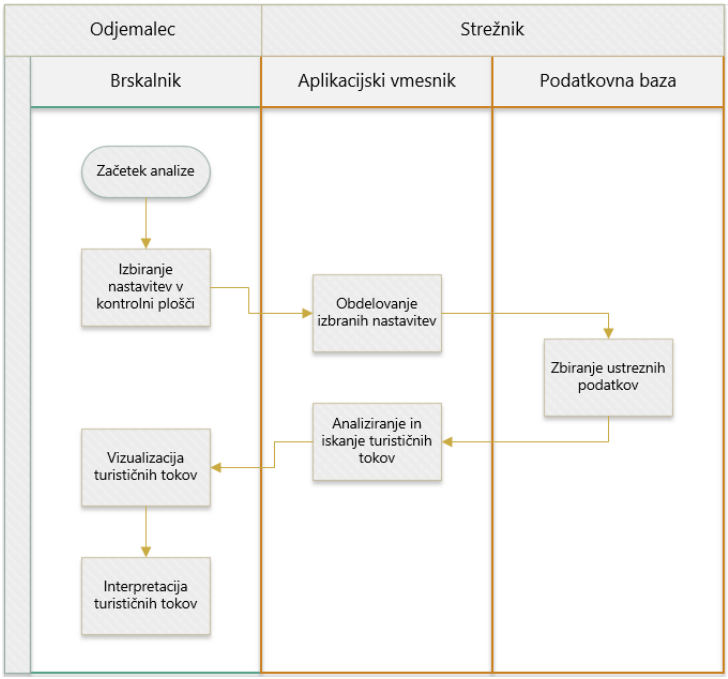
lec. Strežnik vsebuje podatkovno bazo (*Podatkovna baza*), ki nudi podatke aplikacijskemu vmesniku (*API*). Ta iz podatkovne baze vzame in sprocesira, po metodologiji iz poglavja 5, podatke, ki jih nato preko vmesnika nudi odjemalcu.

Odjemalca, ki je v našem primeru spletni brskalnik, sestavljajo tri komponente. Kompontenta - *Kontrolna plošča* nudi uporabniku možnost izbire načina analize in izbire raznih kriterijev ter nastavitev vizualizacije. Za prikaz vizualizacije uporabljamo zemljevid (*Vizualizacija*). Za bolj podrobno

analizo posameznih turističnih točk in turističnih tokov pa služi komponenta -*Podrobna analiza*.

6.1.4 Potek analize

Glavni namen našega prototipa je analiza turističnih tokov. Za identifikacijo le-teh uporabljamo metodologijo, ki je predstavljena v poglavju 5. Za podrobnejši prikaz sodelovanja posameznih komponent, smo uporabili diagram aktivnosti. Diagram aktivnosti je namenjen grafičnem prikazu poteka procesa in sodelovanju z drugimi procesi ali komponentami. Na sliki 6.4, je prikazan primer uporabe za vizualizacijo turističnih tokov. Uporabnik mora



Slika 6.4: Diagram aktivnosti analize turističnih tokov.

v brskalniku za analizo najprej izbrati ustrezne nastavitve. Ob potrditvi spletni brskalnik pošlje zahtevo REST na strežnik, kjer aplikacijski vmesnik zahtevo in nastavitve obdela ter iz podatkovne baze pridobi ustrezne podatke. V našem diagramu aplikacijski vmesnik ustvari množico turističnih

tokov po postopku opisanem v podpoglavju 5.2. Aplikacijski vmesnik to isto množico tokov posreduje nazaj brskalniku, ki nato izvede vizualizacijo in jo prikaže končnemu uporabniku. Uporabnik nato lahko s pomočjo vizualizacije in analize interpretira turistične tokove.

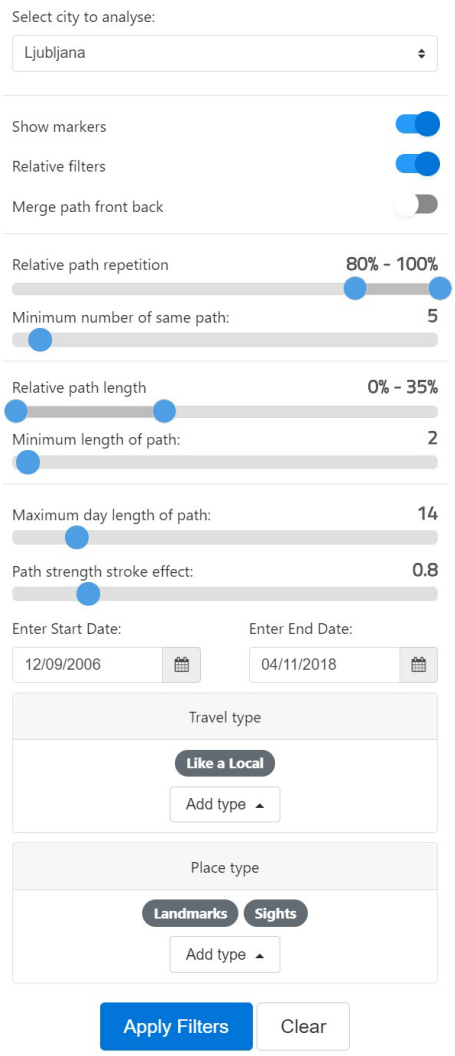
6.2 Izgled in uporaba prototipa

Struktura uporabniškega vmesnika je narejena v označevalnem jeziku HTML, izgled pa v programskem jeziku CSS, s podporo Bootstrapa - za lažje razvrščanje posameznih komponent ter elementov. Bootstrap je prosto dostopna knjižnica, ki nudi ogrodje za izdelavo spletnih strani in na splošno spletnih aplikacij [2]. V osnovi se vidni del prototipa deli na dva glavna dela. To sta kontrola plošča in zemljevid. Kontrolna plošča omogoča nastavljanje posameznih filtrov za analizo, zemljevid pa je namenjen vizualizaciji turističnih tokov, ki ustrezajo izbranim filtrom kontrolne plošče. Za vsak turistični tok in vsako turistično točko so izračunani in ob kliku tudi prikazani posamezni histogrami o spolu, starosti ter tipu turista ali kraja.

6.2.1 Kontrolna plošča

Podatke lahko najprej omejimo z izbiro mesta, v našem primeru lahko izbiramo med mestoma Ljubljana in Dunaj. Za preglednejšo analizo imamo opcijo, da na zemljevidu bodisi skrijemo bodisi prikažemo turistične točke. Za analize pri katerih smer poti ni relevantna, pa lahko izberemo opcijo, kjer obe smeri interpretiramo kot zgolj eno. Izberemo lahko tudi minimalno dolžino turističnega toka ter minimalno število ponovitev poti, ki se zgodijo v enem turističnem toku. Obe vrednosti lahko določimo absolutno ali relativno. Za lažje razločevanje moči turističnih tokov smo dodali tudi možnost izbire odebeljevanja tokov. S tem lahko takoj opazimo najmočnejše turistične tokove pri sami vizualizaciji. Pot, ki jo nek turist opravi, lahko tudi časovno omejimo, kar pomeni, da celotno pot razdelimo na več krajših, ki jih je turist v določenem časovnem območju opravil. Način delitve poti je

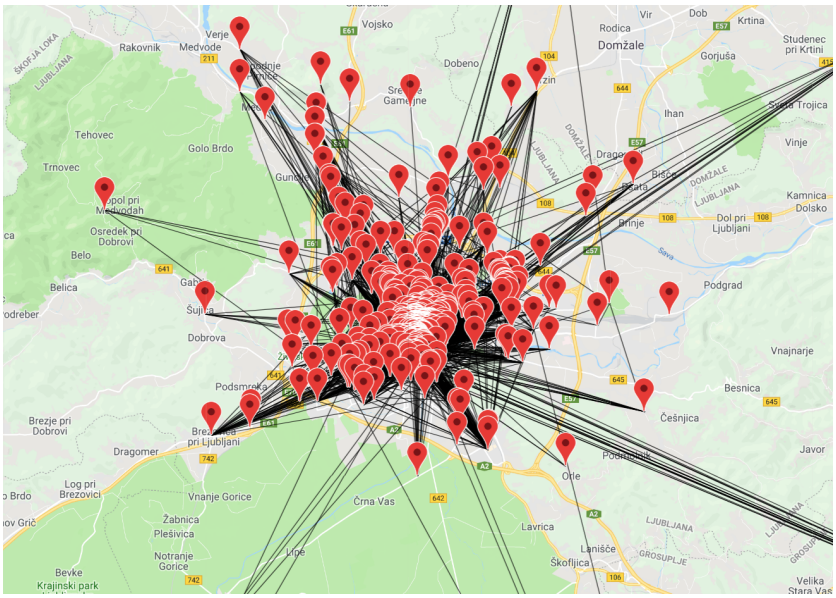
predstavljen v podpoglavju 5.2.3. Vse podatke lahko omejimo tudi z izbiro časovnega obdobja, kjer bomo izvajali analizo. To dosežemo z izbiro datuma začetka ter konca podatkov. Pri analizi turističnih tokov, lahko specifične lastnosti turistov tudi določimo in izberemo. Enako lahko storimo tudi z izbiro lastnosti za turistične točke. Vpliv izbire lastnosti je predstavljen v podpoglavju 5.2.4. Zaslonska slika kontrolne plošča je prikazana na sliki 6.5.



Slika 6.5: Zaslonska slika kontrolne plošče.

6.2.2 Vizualizacija na zemljevidu

Zemljevid smo v prototipu uporabili za prikaz turističnih tokov in turističnih točk. Kot zemljevid smo uporabili prosto dostopno knjižnico, Google maps. Turistične lokacije, ki so jih turisti obiskali in so del analize, na zemljevidu prikažemo s točkami oziroma markerji. V knjižnici se točke imenujejo Google Markers, turistični tok pa je prikazan s črto, ki povezuje turistične točke na zemljevidu. Ta funkcionalnost (črta) se v uporabljeni knjižnici imenuje Google PolyLines. Ob izbiri v kontrolni plošči lahko za lažje ločevanje posameznih moči turističnih tokov, močnejše izrišemo debeleje. Moč turističnega toka v našem primeru označuje število ponovitev iste poti, ki so jo opravili različni turisti. Na sliki 6.6 je prikazana zaslonska slika vizualizacije vseh turističnih tokov in turističnih točk v mestu Ljubljana.



Slika 6.6: Zemljevid in prikaz vizualizacije.

6.2.3 Podrobnejša analiza

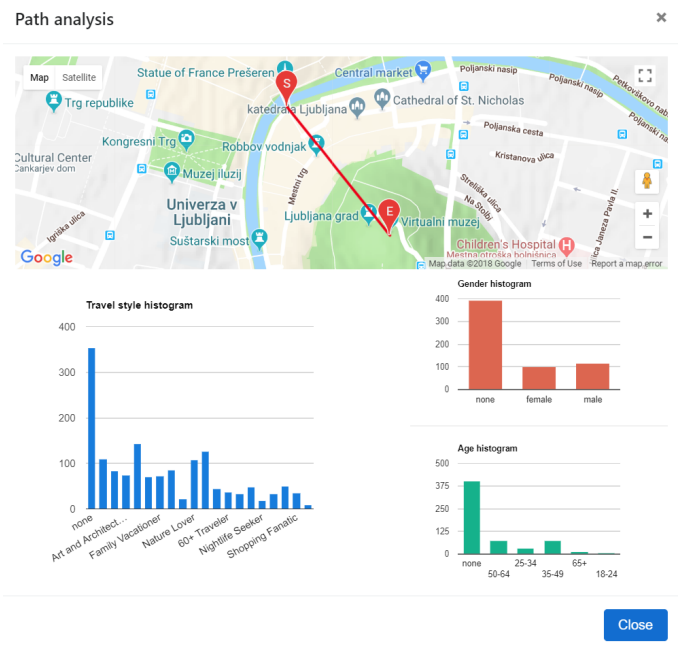
Podatki, ki jih pri analizi uporabljamo, vsebujejo meta podatke o posameznem turistu in lokaciji. Pojem meta podatki ali podatki o podatkih je preširok za neko kratko in jedrnato definicijo, vendar je njihova uporaba prisotna na veliko področjih [9]. Turista lahko opišemo podrobneje s podatki o spolu, starosti in njegovem načinu potovanja, turistično točko pa lahko predstavimo z njeno povprečno oceno, številom obiskov ter vrsto kraja (npr. znamenitost, grad ali muzej). Sami smo te podatke izkoristili za podrobno analizo posameznih turističnih tokov in turističnih točk.

Analiza turističnega toka

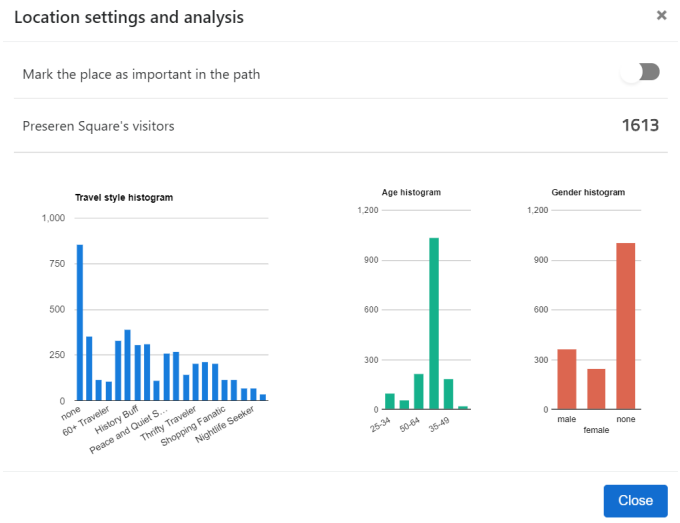
Moč turističnega toka se, kot smo že nekajkrat omenili, meri v številu ponovitev iste poti. Kot osnovo analize turističnega toka smo vzeli meta podatke turistov in iz njih zgradili histograme o spolu, starosti ter tipih turistov, ki so obiskali to pot. Način iskanja specifične poti je predstavljen v podpoglavju 5.2.4. Do teh informacij lahko pridemo s preprostim klikom na turistični tok, ki je izrisan na zemljevidu. Ob kliku se nam odpre novo okno z izrisano potjo, označeno smerjo ter izrisanimi histogrami. Primer analize turističnega toka lahko vidimo na sliki 6.7.

Analiza turistične točke

Turistična točka se za razliko od turističnega toka meri v številu obiskov. Osnova so prav tako turisti, ki so obiskali turistično točko. Ob kliku na točko se izrišejo histogrami o spolu, starosti ter tipu turista. Podana je tudi informacija, kolikokrat je bila točka obiskana. To točko lahko označimo kot pomembno, kar pomeni, da se pri naslednjih analizah upoštevajo zgolj turistični tokovi, ki imajo vsebovano to (pomembno) lokacijo. Primer analize turistične točke lahko vidimo na sliki 6.8.



Slika 6.7: Analiza turističnega toka (Tromostovje - Ljubljanski grad).



Slika 6.8: Analiza turistične točke (Tromostovje).

Poglavje 7

Rezultati analize

V tem poglavju je predstavljena analiza turističnih tokov na nivoju mesta Ljubljane in Dunaja. Najprej so predstavljeni osnovni rezultati - število različnih turističnih točk in različnih turističnih tokov z upoštevanjem oziroma neupoštevanjem smeri toka. V nadaljevanju je za vsako mesto prikazana analiza turističnih tokov, ki se ponovijo vsaj 20-krat. Pri tej analizi je 15 najmočnejših tokov predstavljenih v tabeli. Nato za vsako mesto podrobneje predstavimo najmočnejši turistični tok, sledi predstavitev najmočnejšega turističnega toka sestavljenega iz vsaj treh različnih lokacij in na koncu še turistični tok, ki smo ga detektirali pri največjem deležu mladih turistov.

7.1 Uporabljena analiza

Analiza turističnih tokov, katere smo se poslužili v našem prototipu, temelji na nivoju mesta. To pomeni, da je analiza izvedena na podlagi posameznih turističnih točk oziroma lokacij, ki smo jih zajeli iz spletnega mesta TripAdvisor. V diplomski nalogi smo analizirali območje glavnega mesta Slovenije - Ljubljano in Avstrije - Dunaj. Turistični tok z vidika turista predstavlja njegovo premikanje v izbranem časovnem obdobju znotraj posameznega glavnega mesta, v našem primeru Ljubljane ali Dunaja.

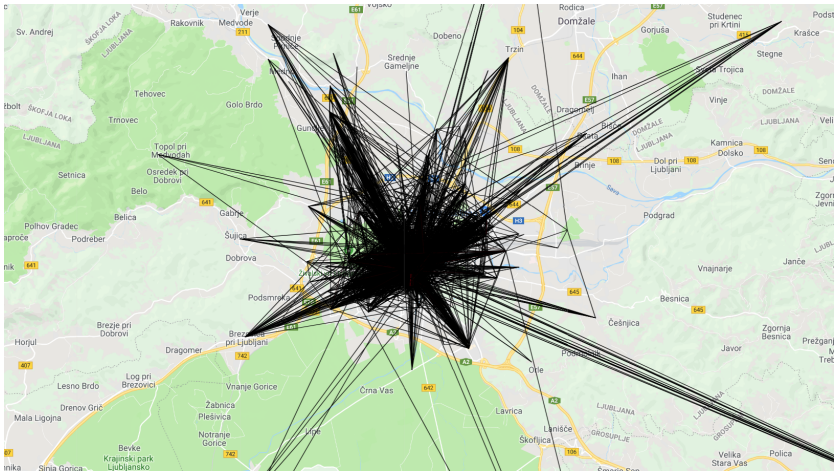
Nastavitve oziroma parametri, ki smo jih uporabili pri analizi Ljubljane

ter Dunaja so identične. Za časovno območje, ki določa dovoljen čas med objavami posameznega turista, smo si izbrali obdobje štirinajstih dni. To pomeni, da je množica obiskov, ki med seboj časovno niso oddaljeni več kot štirinajst dni, predstavljena kot ena pot. Podrobneje opisani časovni intervali ter gradnje poti, so predstavljeni v podpoglavju 5.2.3. Podatki, ki smo jih uporabili za analizo pa so nastali v časovnem obdobju od začetka leta 2005 do začetka leta 2018.

Že pred analizo je moč pričakovati, da bo delež krajših turističnih tokov precej večji od daljših. Razlog temelji predvsem na majhnem deležu turistov, ki v kratkem časovnem obdobju objavijo veliko mnenj. V sklopu identifikacije turističnih tokov to interpretiramo kot posamezno pot. Med drugim lahko pričakujemo tudi, da bo delež turističnih tokov z manj ponovitvami večji. To se lahko zgodi kot posledica analize na nivoju posameznih lokacij znotraj mesta.

7.2 Analiza Ljubljane

Mesto Ljubljana vsebuje, z našimi podatki, 565 različnih lokacij. Kot že omenjeno (poglavje 5) v sklopu analize Ljubljane, poleg objav o atrakcijah, upoštevamo tudi objave o restavracijah, kar za Dunaj ne velja. Na podlagi analize smo torej identificirali 9938 različnih turističnih tokov. Posamezni turistični tokovi so bili obiskani tudi v obratni smeri. Smer poti, v našem primeru turističnega toka, v splošnem predstavlja zaporedje obiska posameznih lokacij. Podrobnejša razlaga pomena ter vpliva smeri, je predstavljena že v podpoglavju 5.2.3, kjer je tudi opisana uporabljena metodologija detekcije turističnih tokov. V primeru brez upoštevanja smeri, kjer smo tokove ne glede na njihovo usmerjenost interpretirali kot enake, pa smo identificirali 9270 različnih turističnih tokov. Na sliki 7.1 so prikazani vsi turistični tokovi, ki smo jih identificirali v mestu Ljubljana. Ogromen delež turističnih tokov v Ljubljani ima, kot smo pričakovali, samo eno ponovitev, kar pomeni, da je to pot opravil zgolj en turist. Takšni turistični tokovi pa v sklopu naše



Slika 7.1: Vsi turistični tokovi v mestu Ljubljana.

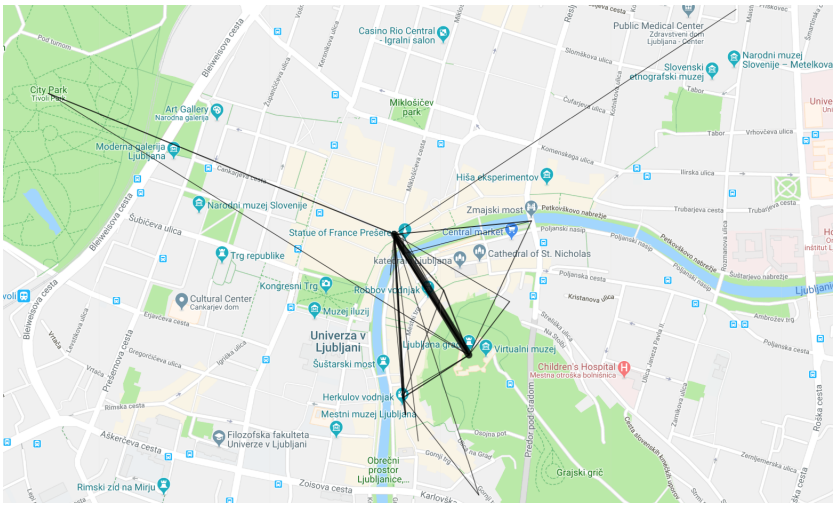
analize ne pridejo v poštev. Teh tokov je izmed vseh največ, približno 89%. Delež turističnih tokov, ki imajo natanko dve ponovitvi, je zgolj 6%. Največji delež turističnih tokov so tokovi, ki so sestavljeni iz obiskov dveh različnih lokacij. Teh je namreč 33%. Sledijo jim turistični tokovi, ki so sestavljeni iz treh različnih lokacij (27%). Kot je bilo pričakovati, je delež krajših turističnih tokov večji. Na sliki 7.1 lahko tudi opazimo, da največ turističnih tokov poteka skozi center mesta.

Sledi predstavitev najmočnejših turističnih tokov, identificiranih v Ljubljani. Najbolj zanimivi izmed njih pa so kasneje še podrobneje predstavljeni.

7.2.1 Najmočnejši turistični tokovi Ljubljane

Tokovi, s pomočjo katerih lahko izluščimo čim več koristnih informacij in so v končni fazi tudi zanimivi, potrebujejo čim večje število ponovitev. S tem namenom smo pri analizi upoštevali zgolj tokove, ki imajo vsaj 20 ponovitev, kar pomeni, da je moralo vsaj 20 različnih turistov objaviti komentarje, na spletnem mestu TripAdvisor, v enakem vrstnem redu in brez večjih časovnih razmikov. Turističnih tokov, ki ustrezajo izbranim zahtevam, je v našem primeru 35, z upoštevanjem smeri pa le 24. Tisto območje, kjer se omenjeni

turistični tokovi zgostijo, je znotraj mesta. Zgoščeni tokovi tako ustvarijo navidezni trikotnik med lokacijami: Prešernov trg, Ljubljanski grad ter kopico restavracij v okolici Gallusovega nabrežja. Ugotovljeno lahko podpremo s sliko 7.2, kjer vidimo, da vsaj 20-krat ponovoljeni turistični tokovi, tvorijo omenjeni navidezni trikotnik. Največji delež turističnih tokov znotraj mesta



Slika 7.2: Najmočnejši tokovi v mestu Ljubljana.

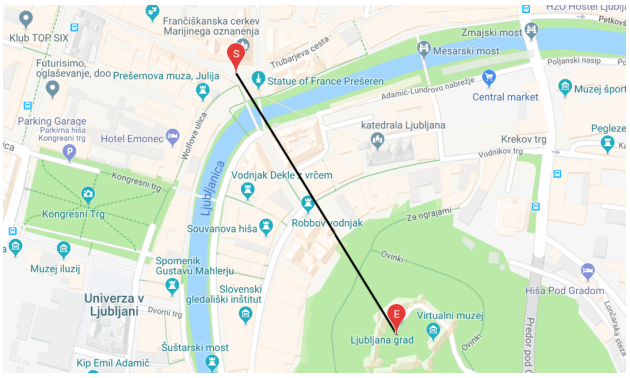
je sestavljenih iz dveh turističnih točk, kar pomeni, da je turist v času svojega izleta, obiskal dve različni lokaciji. Pojavi se tudi turistični tok, ki ga sestavljajo tri turistične točke. Ti tokovi so pravzaprav pričakovani in predvsem potrjujejo, da uporabljen pristop deluje pravilno. V nadaljevanju analiza z vsebinskega stališča postane bolj zanimiva, ko z uporabo filtrov - filtri so predstavljeni v podpoglavju 5.2.4 - analiziramo in predstavimo najmočnejše turistične tokove.

7.2.2 Predstavitev posameznih tokov

V tabeli 7.1 je predstavljenih 15 najmočnejših turističnih tokov v Ljubljani. Tabela vsebuje identifikator posameznega toka. Hkrati vsebuje tudi število, ki prikazuje vse ponovitve poti, brez upoštevanja smeri. Predstavljena je tudi

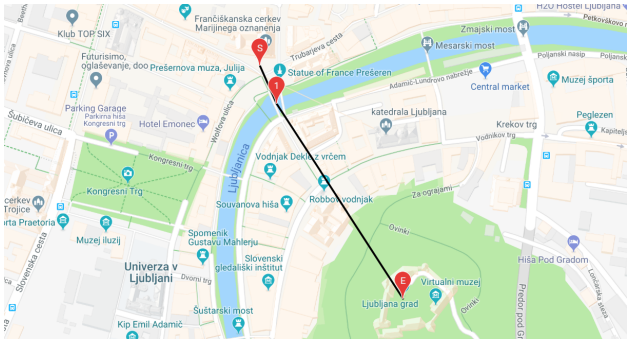
dolžina poti ter glavna pot, kateri je pripisano število ponovitev. Enako je predstavljena tudi njena obratna smer. V tabeli sta med drugim prikazana tudi deleža obiskovalcev turističnega toka glede na spol in starost. Pri tem sta pri starosti predstavljeni le dve najvišji vrednosti.

V Ljubljani ima najmočnejši turistični tok 286 ponovitev in poteka iz starega mesta Ljubljane na Ljubljanski grad. Če ta tok razdelimo na smer, se omenjena smer ponovi 187-krat, njegova obratna pa le 99-krat. Turistični tok smo z 59% deležem identificirali pri turistih moškega spola, od česar s 40% deležem predstavljajo moški stari od 35 let do 49 let. Tok je v tabeli 7.1 predstavljen na prvem mestu in je prikazan na sliki 7.3.



Slika 7.3: Najmočnejši turistični tok v Ljubljani.

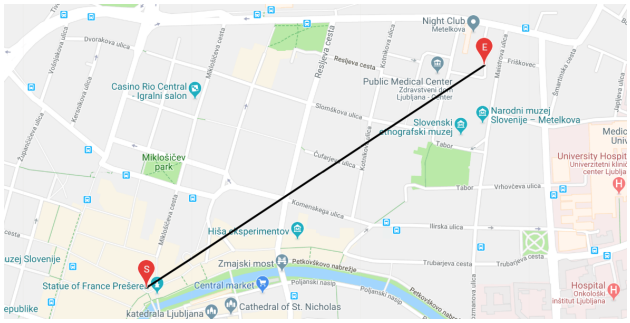
Najmočnejši turistični tok, ki ga sestavljajo tri turistične točke je v tabeli 7.1, označen z zaporedno vrednostjo 9. Ponovi se 45-krat in vsebuje lokacije: staro mesto Ljubljane, Tromostovje ter Ljubljanski grad. Tudi ta tok smo analizirali glede na spol, pri čemer smo ugotovili, da prevladujejo moški z 52% deležem. Tok je prikazan na sliki 7.4 in je na prvi pogled zelo podoben najmočnejšemu toku v Ljubljani. Na tem mestu je smiselno omeniti, da oba toka - tako najmočnejši, kot tudi najmočnejši dolžine tri - vsebujeta točko, ki se imenuje staro mesto Ljubljane. Staro mesto Ljubljane v splošnem ni turistična točka, temveč območje, kar lahko pomeni, da v sklopu



Slika 7.4: Najmočnejši turistični tok dolžine 3 v mestu Ljubljana.

najmočnejšega toka ni nujno, da so turisti sploh obiskali Tromostovje, pri čemer ga v toku dolžine 3 so.

Z analizo smo identificirali turistični tok, ki vsebuje objave predvsem mlajših turistov. V tabeli 7.1 je predstavljen pod zaporedno številko 14. Tok poteka iz centra Ljubljane proti Metelkovi ulici in se skupaj ponovi 36-krat. Če ta tok razdelimo na smer, se že omenjena, ki je med drugim tudi najmočnejša, ponovi 21-krat. Največji delež turistov (67%), pri katerih smo ta tok identificirali, je starih med 25 in 34 let. Iz večernih prireditev na Metelkovi in iz značilnosti lokacije lahko sklepamo, da mlajši turisti najprej obiščejo center Ljubljane ter se nato v večernih urah odpravijo na Metelkovo. Na sliki 7.5 je prikazan omenjen turistični tok.



Slika 7.5: Turistični tok z največjim deležem mladih.

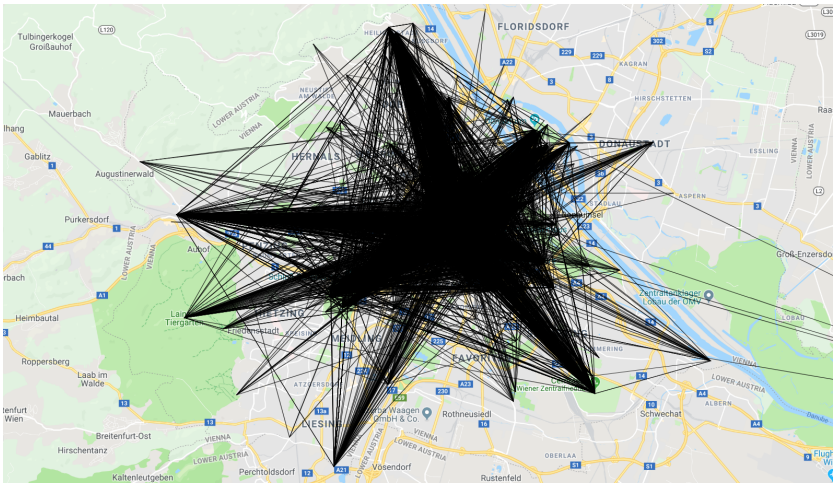
Tabela 7.1: Najmočnejši turistični tokovi v mestu Ljubljana

ID TT	Vse ponovitve	Dolžina	Glavna pot	Obratna pot	M/Ž	Starost 1., 2.
1	286	2	SML-LG=187	LG-SML=99	M=59%, Ž=41%	1. 35-49=40%, 2. 50-64=34%
2	113	2	SML-T=93	T-SML=20	M=31%, Ž=69%	1. 50-64=47%, 2. 35-49=27%
3	81	2	LG-T=53	T-LG=28	M=72%, Ž=28%	1. 25-34=39%, 2. 35-49=26%
4	78	2	rJ-SML=51	SML-rJ=27	M=63%, Ž=37%	1. 35-49=32%, 2. 65+=26%
5	57	2	rMM-rJ=34	rJ-rMM=23	M=48%, Ž=52%	1. 35-49=44%, 2. 50-64=44%
6	52	2	rJ-LG=43	LG-rJ=9	M=73%, Ž=27%	1. 35-49=44%, 2. 50-64=28%
7	52	2	rGSL-LG=36	LG-rGSL=16	M=56%, Ž=44%	1. 35-49=57%, 2. 25-34=21%
8	46	2	rGSL-SML=28	SML-rGSL=18	M=64%, Ž=36%	1. 50-64=40%, 2. 35-49=33%
9	45	3	SML-T-LG=36	LG-T-SML=9	M=52%, Ž=48%	1. 35-49=31%, 2. 50-64=30%
10	43	2	SML-TP=33	TP-SML=10	M=60%, Ž=40%	1. 35-49=43%, 2. 50-64=28%
11	39	2	SML-PT=26	PT-SML=13	M=47%, Ž=53%	1. 50-64=41%, 2. 35-49=29%
12	38	2	rV-rJ=21	rJ-rV=17	M=58%, Ž=42%	1. 50-64=45%, 2. 35-49=27%
13	37	2	SML-ZM=25	ZM-SML=12	M=50%, Ž=50%	1. 25-34=40%, 2. 35-49=30%
14	36	2	LG-M=21	M-LG=15	M=42%, Ž=58%	1. 25-34=67%, 2. 35-49=25%
15	33	2	ZM-LG=24	LG-ZM=9	M=70%, Ž=30%	1. 35-49=33%, 2. 50-64=33%

Kratice lokacij: SML - Staro mesto Ljubljane; LG - Ljubljanski grad; T - Tromostovje; rJ - restavracija Julia; rMM - Restavracija Marley & Me; rGSL - Gostilna Sokol Ljubljana; TP - Tivoli Park; rV - Restavracija Valvasor; PT - Prešernov Trg; M - Metelkova; ZM - Zmajski most

7.3 Analiza Dunaja

Mesto Dunaj sestavlja 555 različnih turističnih lokacij. Kot smo že omenili pri analizi Ljubljane, Dunaj vsebuje zgolj podatke o atrakcijah, ne pa tudi o restavracijah. Z upoštevanjem pridobljenih podatkov za Dunaj, smo identificirali 23678 turističnih tokov z upoštevanjem smeri. Smer je obrazložena že v podpoglavju 5.2.3, kjer je prav tako prikazana metodologija detekcije turističnih tokov. Brez upoštevanja smeri pa smo identificirali 22174 turističnih tokov. Na sliki 7.7, so torej prikazani vsi turistični tokovi v mestu Dunaj. Največji delež turističnih tokov, so tokovi, ki imajo zgolj eno ponovitev. Teh



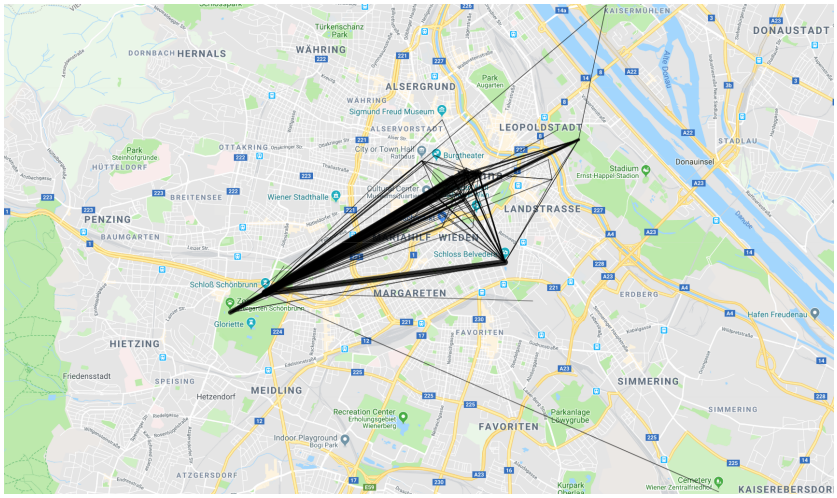
Slika 7.6: Vsi turistični tokovi v mestu Dunaj.

je namreč 88%. Delež turističnih tokov z dvema ponovitvama pa je zgolj 5%. V naši analizi turistični tokovi s tako malo ponovitvami ne pridejo v poštev. Ob upoštevanju vseh turističnih tokov, je največji delež tokov dolžine tri, kar 23%. Večina poti poteka skozi center Dunaja, kar je razvidno že s slike 7.7.

7.3.1 Najmočnejši turistični tokovi Dunaja

Analize Dunaja smo se lotili z enakimi parametri in nastavitvami, kot analize Ljubljane. Analizirali smo torej turistične tokove, ki se ponovijo vsaj 20-krat,

kar pomeni, da je moralo vsaj 20 različnih turistov objaviti svoje mnenje na spletnem mestu TripAdvisor v enakem vrstnem redu in brez večjih časovnih razmikov. Relativni delež turističnih tokov, ki se ponovijo vsaj 20-krat, je manjši od 1%. Turistični tokovi, ki ustrezajo zahtevam, se zgostijo znotraj mesta in ustvarijo navidezen trikotnik med centrom mesta, znanim Dunajskim parkom Schönbrunn ter palače Belvedere. Omenjeni trikotnik vidimo na sliki 7.7. Turističnih tokov je torej brez upoštevanja smeri, v našem pri-



Slika 7.7: Najmočnejši tokovi v mestu Dunaj.

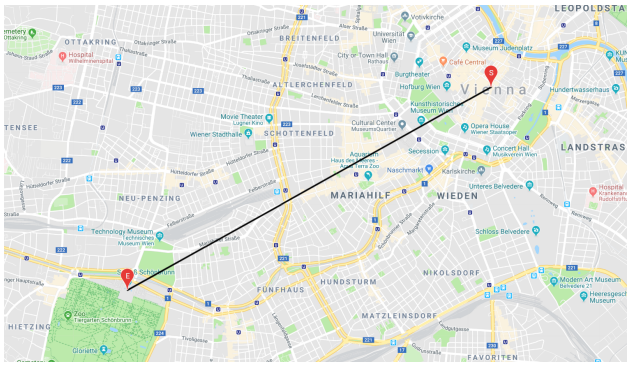
meru 187, upoštevajoč smer pa 171. Turistični tokovi dolžine 2, prevladujejo s 73% deležem. Sledijo jim tokovi dolžine 3, s 27%. Iz omenjenih vrednosti lahko torej razberemo, da se tokovi ostalih dolžin ponovijo manj kot 20-krat.

Nadaljnje so podrobneje predstavljeni najbolj zanimivi posamezni turistični tokovi.

7.3.2 Predstavitev posameznih tokov

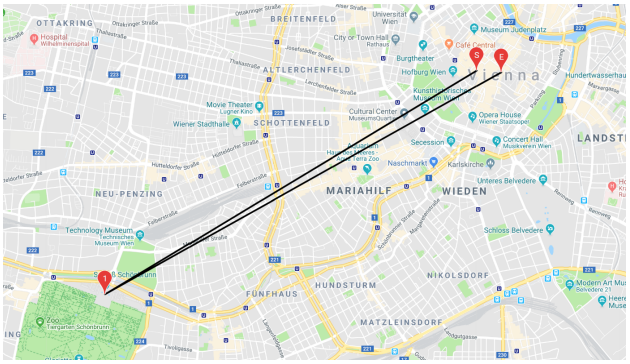
V tabeli 7.2 je prikazanih 15 najmočnejših turističnih tokov, ki smo jih identificirali v mestu Dunaj. Posamezne komponente tabele so opisane že v poglavju 7.2.2, kjer predstavimo posamezne najmočnejše turistične tokove v Ljubljani.

Najmočnejši turistični tok se v Dunaju ponovi 923-krat in poteka iz zgodovinskega centra Dunaja do palače Schönbrunn. Omenjena smer toka se ponovi 533-krat, pri čemer se njegova obratna zgolj 390-krat. Tok smo s 56% deležem identificirali pri turistih moškega spola, od česar s 36% deležem predstaljajo moški stari od 50 let do 64 let. Tok je v tabeli 7.2 predstavljen z zaporedno vrednostjo 1 in je prikazan na sliki 7.8.



Slika 7.8: Najmočnejši turistični tok v mestu Dunaj.

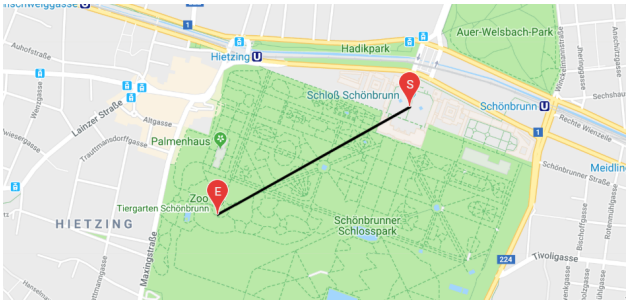
Turistični tok, ki je sestavljen iz treh različnih turističnih točk, se skupaj ponovi 109-krat in je v tabeli 7.2 predstavljen pod zaporedno številko 15. Tok poteka iz zgodovinskega centra mesta Dunaj proti palači Schönbrunn in se konča v katedrali svetega Štefana. Omenjena smer se ponovi 95-krat, kar je precej več od njene obratne smeri, ki ima zgolj 14 ponovitev. Razmerje objav glede na spol, iz katerih smo identificirali tok, prevladuje moški spol s 53%. Tok dolžine tri je prikazan na sliki 7.9. Najmočnejša turistična tokova dolžine 3 in 2 sta si, kot lahko na slikah opazimo, zelo podobna. Tukaj je potrebno omeniti, da je v toku dolžine 3 kot prva lokacija lokacija zgodovinski center mesta Dunaj. Zgodovinski center mesta Dunaj je pravzaprav območje in ne lokacija. Iz tega lahko sklepamo, da si tokova nista povsem enaka, hkrati pa v toku dolžine 3 opazimo, da izstopa smer. Pozanimali smo se, kdaj se ogledi posameznih znamenitosti predvidoma zapirajo. Ugotovili smo, da se palača Schönbrunn zapre nekaj ur pred katedralo svetega Štefana [10, 15]. S pomočjo te informacije lahko sklepamo, da so turisti s tem razlogom najprej



Slika 7.9: Najmočnejši turistični tok dolžine 3 v mestu Dunaj.

obiskali palačo Schönbrunn, šele nato, kot zadnjo, katedralo svetega Štefana.

Turistični tok, ki ga prepotujejo večinoma mladi turisti, se ponovi 527-krat in kot lokacije vsebuje znani živalski vrt na Dunaju ter palačo Schönbrunn. Ob upoštevanju smeri, ima najmočnejša smer 341 ponovitev in se začne v palači Schönbrunn ter konča v živalskem vrtu. Tok je v tabeli 7.2 predstavljen pod zaporedno številko 4 in je prikazan na sliki 7.10.



Slika 7.10: Turistični tok z največjim deležem mladih v mestu Dunaj.

Tabela 7.2: Najmočnejši turistični tokovi v mestu Dunaj

ID TT	Vse ponovitve	Dolžina	Glavna pot	Obratna pot	M/Ž	Starost 1., 2.
1	923	2	ZCD-PS=533	PS-ZCD=390	M=56%, Ž=44%	1. 50-64=36%, 2. 35-49=33%
2	902	2	PS-KSŠ=694	KSŠ-PS=208	M=47%, Ž=53%	1. 35-49=33%, 2. 50-64=32%
3	755	2	PS-MPD=584	MPD-PS=170	M=42%, Ž=58%	1. 35-49=37%, 2. 50-64=31%
4	527	2	PS-ŽD=341	ŽD-PS=186	M=48%, Ž=52%	1. 25-34=38%, 2. 35-49=38%
5	522	2	PS-HP=408	HP-PS=114	M=35%, Ž=65%	1. 50-64=35%, 2. 35-49=34%
6	374	2	PS-SV=260	SV-PS=114	M=42%, Ž=58%	1. 50-64=36%, 2. 35-49=27%
7	370	2	ZCD-KSŠ=234	KSŠ-ZCD=136	M=55%, Ž=45%	1. 35-49=41%, 2. 50-64=30%
8	361	2	PS-OHD=232	OHD-PS=129	M=49%, Ž=51%	1. 35-49=34%, 2. 50-64=28%
9	356	2	PS-P=239	P-PS=117	M=52%, Ž=48%	1. 35-49=39%, 2. 50-64=20%
10	313	2	PS-DUZ=158	DUZ-PS=155	M=46%, Ž=54%	1. 50-64=46%, 2. 35-49=26%
11	232	2	KSŠ-MPD=122	MPD-KSŠ=110	M=47%, Ž=53%	1. 35-49=44%, 2. 25-34=25%
12	211	2	ZCD-MPD=138	MPD-ZCD=73	M=49%, Ž=51%	1. 50-64=45%, 2. 35-49=38%
13	162	2	HP-KSŠ=82	KSŠ-HP=80	M=54%, Ž=46%	1. 35-49=37%, 2. 50-64=37%
14	138	2	KSŠ-P=93	P-KSŠ=45	M=58%, Ž=42%	1. 50-64=32%, 2. 25-34=30%
15	109	3	ZCD-PS-KSŠ=95	KSŠ-PS-ZCD=14	M=53%, Ž=47%	1. 35-49=32%, 2. 50-64=32%

Kratice lokacij: ZCD - Zgodovinski center Dunaja; PS - Palača Schönbrunn; KSŠ - Katedrala sv. Štefana; MPD - Muzej palače Belvedere; ŽD - Živalski vrt Dunaj; HP - Hofburgkška palača; SV - Schönbrunnerski vrtovi; OHD - Operna hiša Dunaj; P - Prater; DUZ - Dunajski muzej umetnostne zgodovine

Poglavje 8

Sklepne ugotovitve

Cilj diplomskega dela je temeljil na konceptu metodologije, ki je predstavljena v članku o analizi turističnih tokov v Sloveniji [3]. Metodologijo smo preuredili in prilagodili za identifikacijo turističnih tokov znotraj mesta in jo uporabili v prototipu. Abstrakten koncept metodologije je do neke mere ostal podoben, vendar smo zaporedne korake, ki so omenjeni v članku spremenili, ker smo hkrati izvedli tudi analizo turistov, ki pa v članku ni omenjena. Pomembno je bilo tudi upoštevanje vsake lokacije, kar pomeni, da smo izpustili korak združevanja lokacij, ki je prav tako omenjen v članku. Navsezadnje pa smo dodali metodologiji tudi korak filtriranja, ki nam omogoča lažjo in preglednejšo analizo. Diplomsko delo se v splošnem deli na pomembne komponente (oz. korake) prototipa, ki so podrobneje predstavljeni s poglavji.

V sklopu diplomskega dela je bil izdelan prototip, ki identificira turistične tokove znotraj mesta na podlagi spletnih objav turistov. Prototip v splošnem omogoča izbiro raznih parametrov iz kontrolne plošče, za bolj specifično identifikacijo in analizo turističnih tokov. Za lažje razumevanje omenjenih tokov pa je izvedena tudi vizualizacija na zemljevidu. Podatke, ki jih prototip uporablja, smo izluščili iz spletnega mesta TripAdvisor.

Prototip smo v diplomskem delu uporabili in z njim uspešno izvedli analizo nad mestoma Ljubljana in Dunaj, pri čemer smo identificirali ogromno turističnih tokov in najmočnejše tudi preučili ter predstavili. Ugotovili smo,

skozi katere predele mesta gre največ turistov. Te turiste smo tudi opredelili in analizirali. S tem smo dosegli na začetku zadane cilje.

Prototip bi lahko izboljšali predvsem z vidika zajema podatkov. Zanimivo bi bilo na primer zajeti podatke še s kakšnega drugega spletnega mesta, ne le iz spletnega mesta TripAdvisor. Rezultate analize bi lahko tako med seboj primerjali iz več različnih spletnih mest. Možna nadgradnja bi bila lahko tudi z vidika metodologije, kjer bi izboljšali korak deljenja poti na način, da bi pregledovali tudi vsebovanost krajših poti v daljših. S takšno nadgradnjo bi dobili veliko krajših in močnejših turističnih tokov, vendar pa bi vsebinsko tak tok težje interpretirali kot, če bi ga upoštevali zgolj enkrat. Prototip bi lahko nadgradili tudi z vidika analize, kjer ne bi analizirali zgolj turističnih tokov znotraj mesta, temveč bi analizirali način potovanja turistov. Analizirali bi njihove preostale obiske na različnih krajih in te kraje navsezadnje primerjali med seboj. Vsekakor pa bi lahko spletno aplikacijo nadgradili še s kakšno dodatno funkcionalnostjo, ki bi nam analizo podatkov še dodatno olajšala.

Literatura

[1] Sanjay Agrawal, Vivek Narasayya, and Beverly Yang. Integrating vertical and horizontal partitioning into automated physical database design. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 359–370, New York, NY, USA, 2004. ACM.

[2] Bootstrap (front-end framework). Dosegljivo: `https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)`. [Dostopano: 25. 5. 2018].

[3] Ljubica Knezevic Cvelbar, Mojca Mayr, and Damjan Vavpotic. Geographical mapping of visitor flow in tourism: A user-generated content approach. *Tourism Economics*, 2018.

[4] Marc Delisle. *Mastering phpMyAdmin 3.1 for Effective MySQL Management*. Packt, 2009.

[5] Html. Dosegljivo: `https://en.wikipedia.org/wiki/HTML`. [Dostopano: 28. 4. 2018].

[6] Javascript. Dosegljivo: `https://en.wikipedia.org/wiki/JavaScript`. [Dostopano: 27. 4. 2018].

[7] Jetbrains. Dosegljivo: `https://en.wikipedia.org/wiki/JetBrains`. [Dostopano: 29. 4. 2018].

[8] Laura Thomson Luke Welling. *PHP and MySQL Web Development.* Sams, 2003.

[9] Prakash M. Nadkarni. What is metadata? In *Metadata-driven Software Systems in Biomedicine.*, London, 2011. Springer.

[10] Opening times - schönbrunn. Dosegljivo: `https://www.schoenbrunn.at/en/visitor-information/opening-times/`. [Dostopano: 28. 6. 2018].

[11] Aric Pedersen. *CPanel user guide and tutorial : get the most from cPanel with this easy-to-follow guide.* Packt, Birmingham, U.K., 2006.

[12] Php. Dosegljivo: `https://en.wikipedia.org/wiki/PHP`. [Dostopano: 28. 4. 2018].

[13] Sass (stylesheet language). Dosegljivo: `https://en.wikipedia.org/wiki/Sass_(stylesheet_language)`. [Dostopano: 28. 4. 2018].

[14] Scrapy. Dosegljivo: `https://en.wikipedia.org/wiki/Scrapy`. [Dostopano: 6. 5. 2018].

[15] Stephansdom. Dosegljivo: `http://www.stephanskirche.at/index.jsp?langid=2&menuekeyvalue=11`. [Dostopano: 28. 6. 2018].

[16] Yang Sun, Ziming Zhuang, and C. Lee Giles. A large-scale study of robots.txt. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 1123–1124, New York, NY, USA, 2007. ACM.

[17] Steve F. Tyson. *Decode The PHP Codes: A Simple And Easy PHP Tutorial For Beginners With Clear-Cut Details On HTML Basics, PHP Coding And Other PHP Basics So You Can ... PHP Scripts Like An Expert PHP Programmer.* CreateSpace, Paramount, CA, 2011.